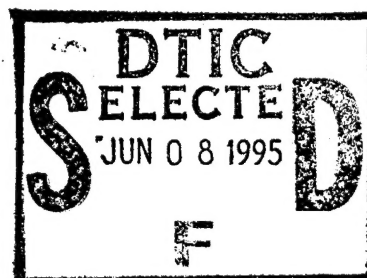# NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

SIMULATION OF A SOLAR POWERED ELECTRIC VEHICLE UNDER THE CONSTRAINTS OF THE WORLD SOLAR CHALLENGE

by

Steven John Roerig

March 1995

Thesis Advisor: Jovan E. Lebaric

Approved for public release; distribution is unlimited

19950607 009

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, Va 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE March 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE
SIMULATION OF A SOLAR POWERED ELECTRIC VEHICLE UNDER THE CONSTRAINTS OF THE WORLD SOLAR CHALLENGE

5. FUNDING NUMBERS

6. AUTHOR(S))
Steven J. Roerig

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, Ca 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Development of an effective method for evaluation of alternative energy sources in the automotive industry has always been a necessity for cost efficient design analysis. One viable alternative energy source is electricity. In the present day environment of shrinking fossil fuel supplies and environmental awareness, electric powered vehicles are becoming a low cost, non-polluting, alternative means of transportation. The analysis of reliable electric propulsion can be expensive without a modeling tool for evaluating design strategies before vehicle construction. This thesis explores electricity as an alternative energy source for the automobile of tomorrow. Under the guidelines of the World Solar Challenge, a solar powered electric vehicle, using a permanent-magnet brushless dc motor has be modeled and simulated in Simulink (Dynamic System Simulation Software). The simulations were performed with the goal of determining the optimum configuration to efficiently utilize the power supplied from the solar array, batteries, and motor. The simulated vehicle was "driven" over various terrain's and at various speeds. The results obtained confirm this simulation as an efficient design tool and present an example of an optimum vehicle speed for one particular vehicle configuration.

14. SUBJECT TERMS
Solar Power, Electric Vehicles, Electric Machines, Simulations

15. NUMBER OF PAGES 103

16 PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std z39-18

i

SIMULATION OF A SOLAR POWERED ELECTRIC VEHICLE UNDER THE CONSTRAINTS OF THE WORLD SOLAR CHALLENGE

by

Steven John Roerig
Lieutenant, United States Navy
B.S., Oregon State University, 1988

Submitted in partial fulfillment of the
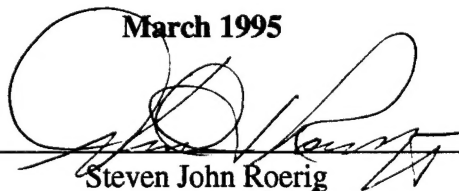requirements for the degree of
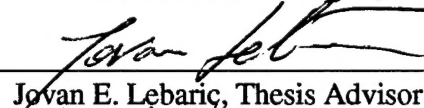
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
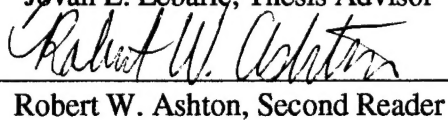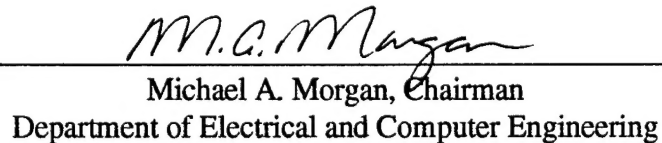
from the

NAVAL POSTGRADUATE SCHOOL

March 1995

Author: _____
Steven John Roerig

Approved by: _____
Jovan E. Lebaric, Thesis Advisor

_____
Robert W. Ashton, Second Reader

_____
Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

Development of an effective method for evaluation of alternative energy sources in the automotive industry has always been a necessity for cost efficient design analysis. One viable alternative energy source is <u>electricity</u>. In the present day environment of shrinking fossil fuel supplies and environmental awareness, electric powered vehicles are becoming a low cost, non-polluting, alternative means of transportation. The analysis of reliable electric propulsion can be expensive without a modeling tool for evaluating design strategies before vehicle construction.

This thesis explores electricity as an alternative energy source for the automobile of tomorrow. Under the guidelines of the World Solar Challenge, a solar powered electric vehicle, using a permanent-magnet brushless dc motor has been modeled and simulated in Simulink (Dynamic System Simulation Software). The simulations were performed with the goal of determining the optimum configuration to efficiently utilize the power supplied from the solar array, batteries, and motor. The simulated vehicle was "driven" over various terrain's and at various speeds. The results obtained confirm this simulation as an efficient design tool and present an example of an optimum vehicle speed for one particular vehicle configuration.

Accesion For		
NTIS CRA&I		☒
DTIC TAB		☐
Unannounced		☐
Justification		
By		
Distribution /		
Availability Codes		
Dist	Avail and / or Special	
A-1		

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS

I. INTRODUCTION

A. BACKGROUND

Development of an effective method for evaluation of alternative energy sources in the automotive industry has always been a necessity for cost efficient design analysis. One viable alternative energy source is <u>electricity</u>. In the present day environment of shrinking fossil fuel supplies and environmental awareness, electric powered vehicles are becoming a low cost, non-polluting, alternative means of transportation. The analysis of reliable electric propulsion can be expensive without a modeling tool for evaluating design strategies before vehicle construction.

The World Solar Challenge is a race that was designed to demonstrate current technology in the field of Solar Powered Electric Vehicles (SPEV). Since it's inception in 1987, it has provided industrial and educational "teams" an opportunity to present the latest technological trends and breakthroughs in solar powered electric vehicle research to the public. The primary means of solar powered vehicle development has been characterized by theoretical evaluation of the vehicles velocity based on four variables: aerodynamic drag, rolling resistance, mass, and power output. Most of the vehicles entered in the World Solar Challenge are evaluated with these four variables before construction.

A more effective design tool would take these variables and incorporate them into a dynamic vehicle model. Coupled with the motor, battery, and solar array, the design engineer could then base the final vehicle design on data accumulated from simulations of this dynamic model. A Solar Powered Electric Vehicle Simulator is therefore absolutely necessary as a primary design tool for cost effective electric vehicle research and development.

B. THESIS OVERVIEW

The primary means for development of solar powered vehicle designs has been to devise a mathematical model based on the performance variables of rolling resistance, air drag, gravity, velocity and required system energy. This model is used to calculate the power required to maintain a continuous speed under steady state conditions. This

approach is adequate for design evaluation when not considering transitions in terrain, speed, or cloud cover.

The goal of this study is to develop a dynamic simulation tool for evaluation of a solar powered electric vehicle model based on the system variables. These system variables will include the vehicle's mechanical and electrical components. This approach will allow the designer to specify and change the vehicle's characteristics in order to optimize the design. In final form, this modeling tool will realistically simulate a solar powered electric vehicle with changing terrain, speed, and solar array power while providing continuous on-line data display.

Figure 1-1 shows a block diagram of the vehicle design considered in this study. The primary components are solar array, battery, voltage/current bus, motor, drive train, and vehicle. Models for the electrical and mechanical system elements are presented independently and tied together to produce the system model.

1. Background of the World Solar Challenge and Solar Powered Vehicles

This section provides a brief history of solar powered electric vehicles and the events surrounding the first World Solar Challenge [Ref. 1, pp. 1-6]. Discussed are the basic rules governing vehicle construction and some of the vehicle designs of previous races.

2. Modeling the Vehicles Electrical Components

Discussed here are the solar array, battery, voltage/current control, and motor. The array and battery characteristics are developed from experimental data and placed into matrices which are used as look-up tables in the system model. The current bus ties together the array current, battery current, and the regenerative current from the motor. The output current form the bus is integrated to monitor the battery *state of charge (SOC)*. The vehicle system model utilizes a permanent-magnet brushless dc motor which is presented in state space form.

3. Modeling the Vehicles Mechanical Components

The system variables which describe the drive train and torque loading characteristics of the vehicle are developed in this section. The forces on the vehicle considered to be of primary importance are rolling resistance, air drag, acceleration force, and hill gravitational force. The coupling equations which describe the drive train are also presented in this section.

2
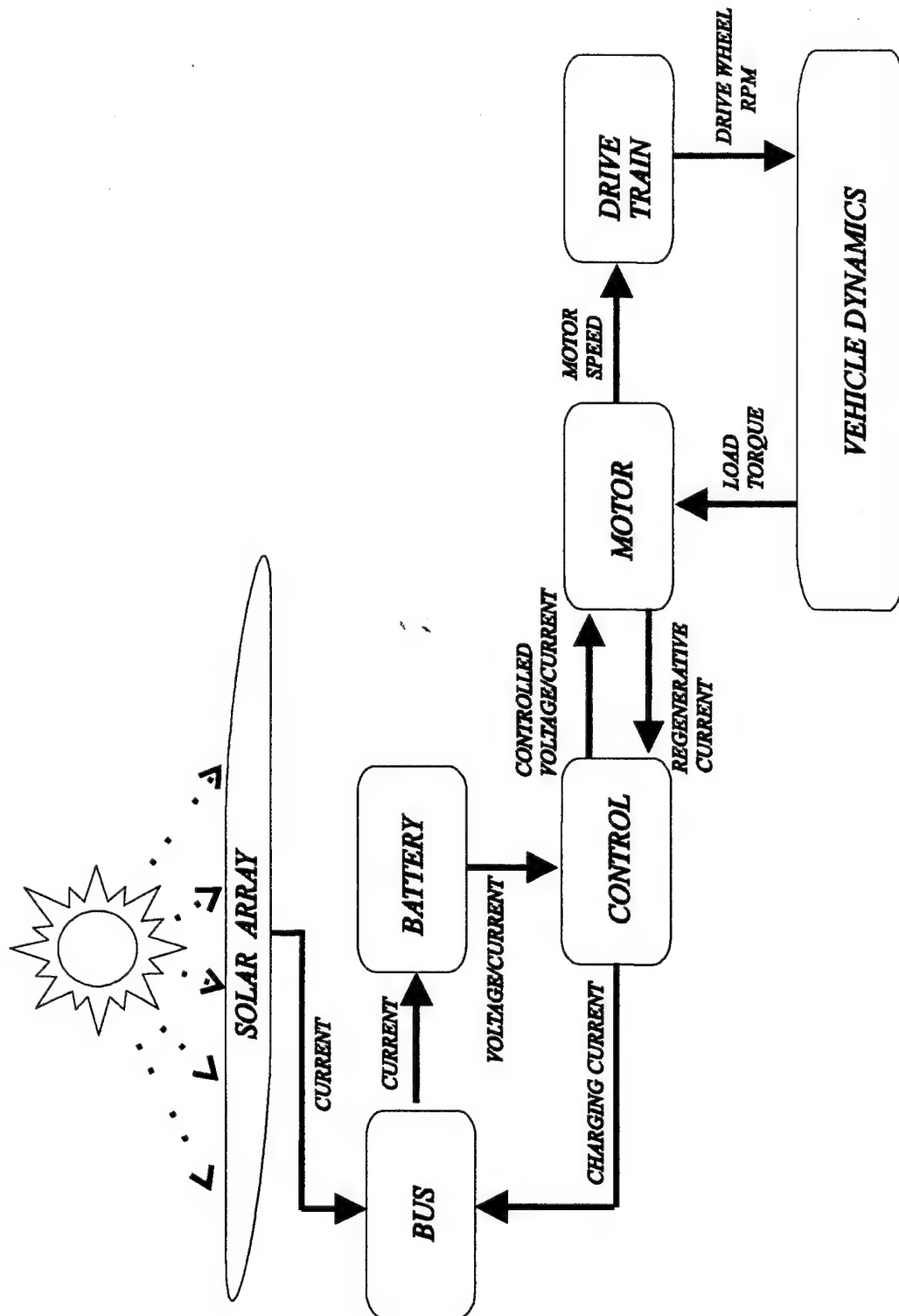
Figure 1-1. Block diagram of solar powered vehicle components to be simulated.

4. Simulation Software and Model Development

Ease of use, speed, and output capabilities are factors considered important when choosing the simulation software. The primary software tools were Matlab and Simulink from MathWorks [Ref. 2]. Both provide for excellent modeling of the linear and non-linear differential equations which form the basic structure of the system. Each vehicle component is developed and presented as a Simulink submodel. These submodels are then tied together to form the dynamic system of the solar powered electric vehicle.

5. Model Response and Data Analysis

The system model is simulated and analyzed for varied driving conditions. Plots of the responses are presented and discussed. The simulation results are compared to the theoretical expected outputs. Then, the simulation is used to present a basic driving strategy to optimize the chosen vehicle configuration.

6. Conclusions and Future Work

Finally, conclusions about the usefulness of the solar powered electric vehicle simulator are presented. Suggestions for expanding the model to include simulation of accessories, power losses, thermal characteristics, and external data accumulation capabilities are presented.

II. BACKGROUND OF THE WORLD SOLAR CHALLENGE AND SOLAR POWERED VEHICLES

A. THE WORLD SOLAR CHALLENGE

1. World Solar Challenge Race Background

The "World Solar Challenge" race was the idea and creation of Hans Tholstrup, the first man to build and drive a solar powered electric vehicle 2500 miles across Australia [Ref. 1, p. 1-6]. He embarked upon that adventure primarily to call attention to the need for a reliable alternative energy source. The World Solar Challenge was an opportunity for private, corporate, and educational teams to prove their various vehicle designs and give the world an opportunity to view the technological advances being made in alternative energy sources for vehicles of the near future.

The first World Solar Challenge brought a field of 24 entrants from around the world. It began on November 1, 1987 and was won by the General Motors team entry *Sunracer*. The average speed of the GM *Sunracer* was 41.8 mph, which allowed the team to cover the 1868 mile distance from Darwin to Adelaide in 44.9 hours.

The design strategies used in the 1987 race set the precedent for the next World Solar Challenge race held in 1990, which brought 36 entrants to Australia. The course was again the 1868 mile Stuart Highway between Darwin and Adelaide. The race was won by the Biel Engineering School team entry *Spirit of Biel II*. The *Spirit of Biel II* averaged 40.5 mph and completed the race 46.1 hours. Table 2-1 shows the vast improvement in the average results achieved by all entrants who completed the race in 1987 and 1990.

2. Rules Governing Vehicle Design

The general rules governing solar powered race vehicle configuration can be used as a basic outline for development of a modeling tool or simulation program. The vehicle limits considered important in this study are:

- maximum solar collector array dimensions are 2 m wide by 4 m long by 1.6 m high - this limits the array power
- maximum vehicle dimensions cannot be greater than 6 m long by 2 m wide 1.6 m high + 1% allowance
- minimum height of the car is 1 m
- minimum weight of driver is 80 kg

These limits provide a basis for design of the submodels developed in the following chapters. Since solar powered electric vehicles are historically designed to be lightweight, the need to optimize the power delivered from the solar array is paramount.

	Average Speed, mph	Average Finishing Time, h
1987	19	124.4
1990	28	69.8

Table 2-1. Comparison of World Solar Challenge Race averages.

III. MODELING THE VEHICLES ELECTRICAL COMPONENTS

A. THE SOLAR ARRAY COMPONENT

The data for a solar array output to an ideal constant voltage bus was taken from data collection experiments performed at Rose-Hulman Institute of Technology [Ref. 3]. The data collection was completed by testing three types of solar array configurations: flat array, curved array, and flat tracking array. The following assumptions and conditions were present during data collection:

- the size of the array conforms to the World Solar Challenge race rules which state the array must fit into a box 1.6 by 2 by 4 meters.
- racing is done between 9 am and 6 pm.
- the day of solar collection is clear and sunny.

Open circuit voltages and short circuit currents were collected every 30 minutes, and the available power was then calculated for each array configuration. Since the simulation uses the current output from an array for a specific time of day, the data may be programmed into a look-up table for use in the simulation routine. Figure 3-1 presents the normalized current output from the three array configurations versus the time of day. Although this data is not exact it does present an approximation that is quite close to the expected results as compared to the array output of the vehicles entered in the 1990 World Solar Challenge [Ref. 1, App. 4]. Figure 3-2 shows the block diagram implementation of the current supplied from the solar array to the vehicle system.

B. THE BATTERY AND BUS COMPONENTS

The battery component of the simulation model is presented in the same form as the solar array. Battery discharge data supplied by the manufacturer [Ref. 1, pp. 73-75] is placed into a matrix and then incorporated into a look-up table for use in the simulation. The bus combines the solar array and battery for output to the motor, and takes regenerative current from the motor to the battery for charging.

1. Modeling the Batteries and Bus

Two types of batteries are used in the model, Lead Acid and Silver Zinc. Figure 3-3 shows the battery voltage versus percent charge or *state of charge (SOC)*. The output voltage to the motor will have a maximum value of 100 volts, although the input to the voltage bus may be larger.
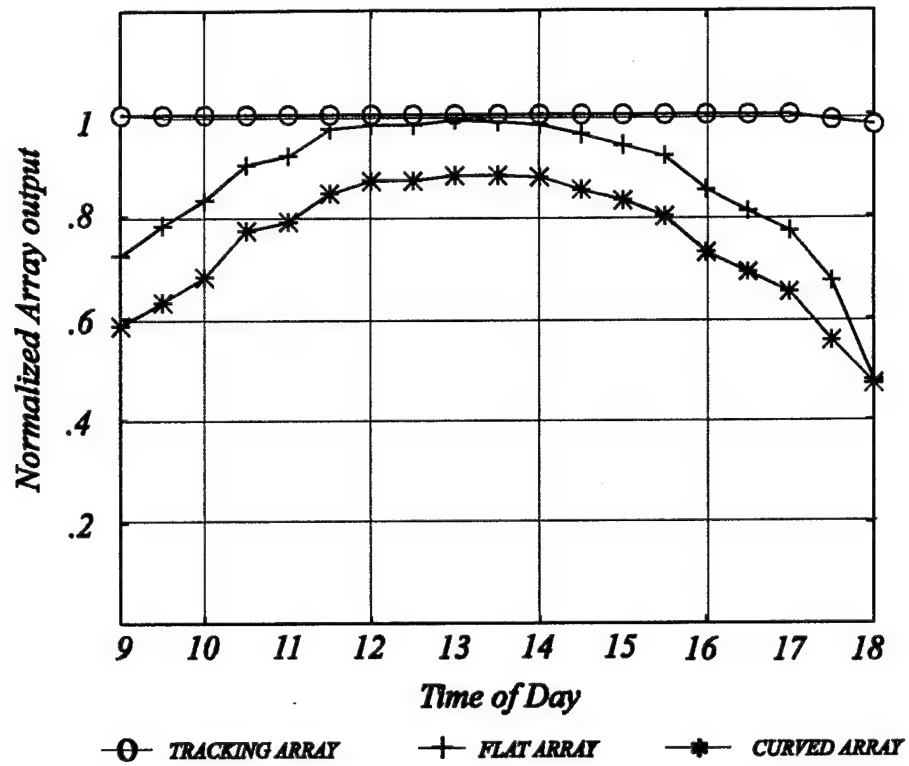
Figure 3-1. Available Current from Solar Array vs. Time of Day.
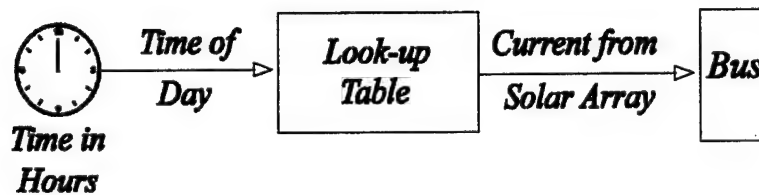
Figure 3-2. Block diagram of current supplied from the solar array to the vehicle system.

8

The battery State-of-Charge model *(SOC(t))* is developed from the Martin Model [Ref. 4, p. 87]. It is calculated as a function of time from the current drawn by the motor,
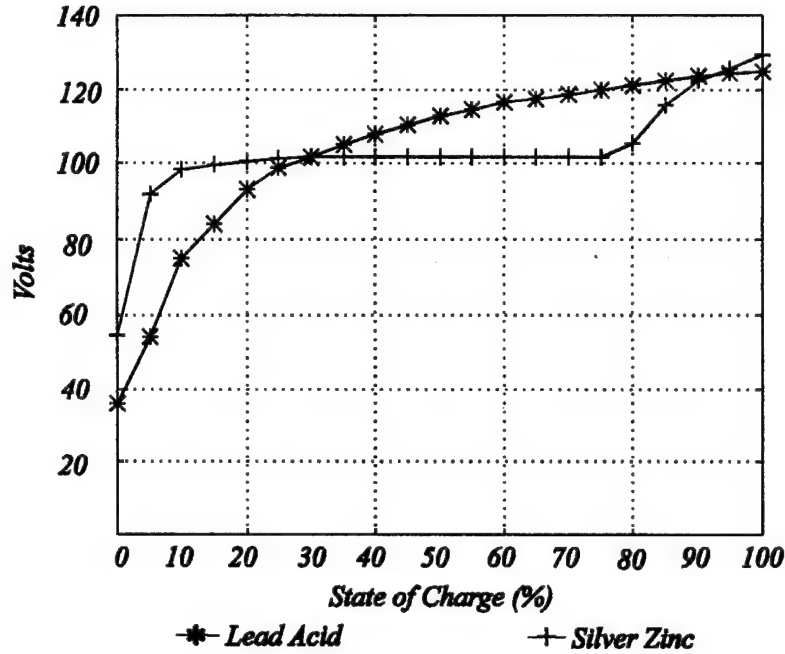
Figure 3-3. Battery Voltage vs. State of Charge.

current supplied by the array, and current regenerated by the motor. An expression for this relationship is

$$SOC(t) = SOC(0) - 100\left(\frac{\int_0^t (i_{mot} - i_{sa})\,dt}{Q_o}\right) \tag{3.1}$$

where Q_o is defined as the battery's maximum current capacity in amp-hours, i_{sa} is the current supplied from the solar array as defined in the previous section, and i_{mot} equals the motor current. The motor current is positive when the motor is drawing current and negative when the motor is generating current. This *regenerative* action of the motor is caused by slowing, stopping, or going down a hill. *SOC(0)* is the battery's initial state of charge (in percent).

9

The block diagram shown in Figure 3-4 presents the implementation of the battery and the voltage bus to be used in the overall vehicle model.
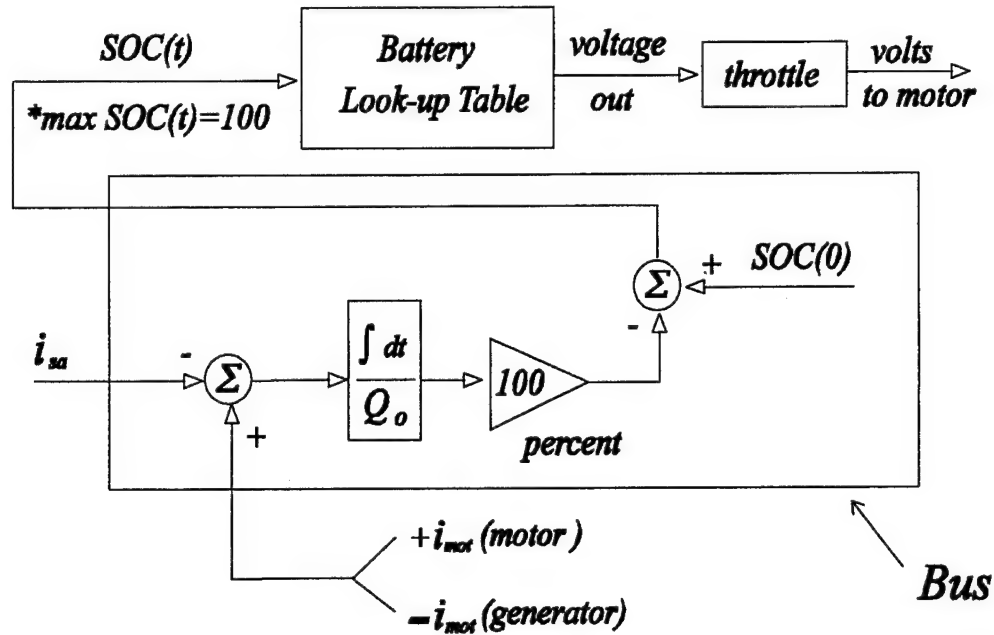
Figure 3-4. Block diagram of battery and bus model to be simulated.

C. THE PERMANENT-MAGNET BRUSHLESS DIRECT CURRENT (PMBDC) MACHINE

A simplified approach to development of a permanent-magnet brushless dc (PMBDC) motor will be presented from the Krause method [Ref. 5, pp. 273-328] of machine analysis. This established theory of PMBDC machines will present the machine dynamic characteristics in terms of linear differential equations, which will then be given in state space form. The matrix of state equations will be the basis for construction of the time-domain block diagram for the dynamic model.

1. The Machine Voltage and Torque Equations

The two-pole two-phase permanent-magnet dc machine, as shown in Figure 3-5, will be used to develop the machine equations. In the circuit diagram, N_s represents the identically distributed *as* and *bs* windings on the stator and r_a the winding resistance. The d axis denotes the rotor magnetic axis, and the q axis measures the angular displacement

10
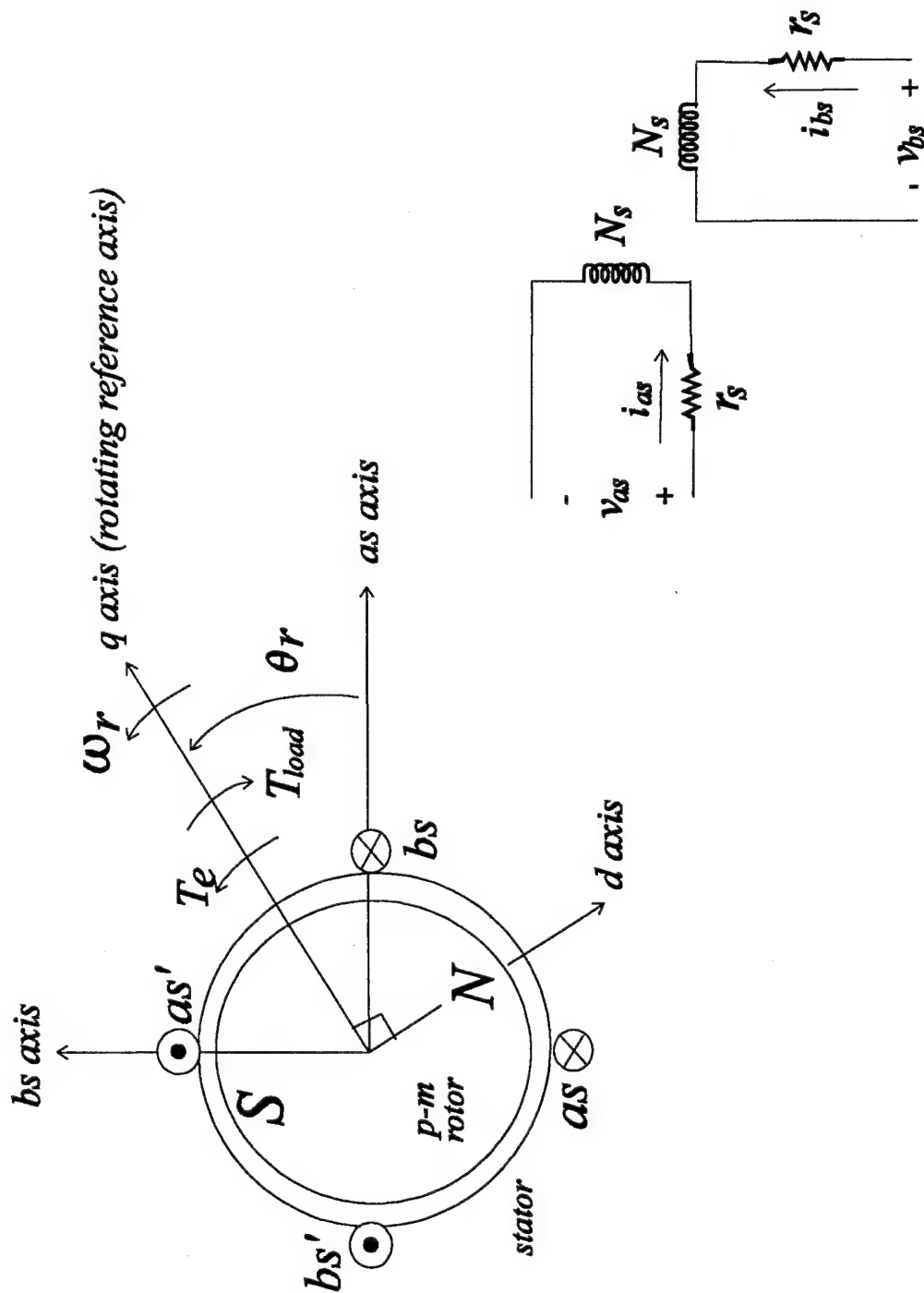
Figure 3-5. Two-pole, two-phase permanent-magnet brushless dc motor.

of the rotor referenced from the stator *as* winding. The *q* axis is displaced 90° counter-clockwise from the *d* axis. The rotating magnetic field produced by the interaction of the stator current and the magnetic rotor develops a positive *electromagnetic torque (T_e)*, which turns the rotor counter-clockwise in the direction of increasing θ_r at an electrical speed of ω_r. The *load torque (T_{load})* acts to oppose the electromagnetic torque.

The following assumptions will be made in our model development:

- No demagnetization of the permanent magnet due to large currents or temperature.
- Eddy currents and harmonic losses are neglected.

A loop analysis of the circuit shown in Figure 3-5 provides the motor voltage equations

$$v_{as} = r_s i_{as} + p\lambda_{as} \tag{3.2}$$

$$v_{bs} = r_s i_{bs} + p\lambda_{bs} \tag{3.3}$$

where *p* is the notation used for the operator *d/dt* and λ is referred to as the *flux linkage*. The sinusoidal voltages v_{as} and v_{bs} are 90° phase shifted. The flux linkage is a function of the inductance of the stator windings and the magnetic flux produced by the permanent-magnet rotor. The flux linkage equations may be expressed as

$$\lambda_{as} = L_{asas} i_{as} + L_{asbs} i_{bs} + \lambda_{asm,\ magnetic\ rotor} \tag{3.4}$$

$$\lambda_{bs} = L_{bsas} i_{as} + L_{bsbs} i_{bs} + \lambda_{bsm,\ magnetic\ rotor} \tag{3.5}$$

and

$$\begin{bmatrix} \lambda_{asm} \\ \lambda_{bsm} \end{bmatrix} = \lambda_m' \begin{bmatrix} \sin\theta_r \\ -\cos\theta_r \end{bmatrix} \tag{3.6}$$

where λ_m' is the amplitude of the flux linkages as seen from the stator, θ_r is defined as

$$\theta_r = \int_0^t \omega_r(\xi)\,d\xi + \theta_r(0) \tag{3.7}$$

where ω_r is the electrical angular velocity and ξ is a dummy variable of integration.

By assuming a uniform air gap and identical windings, the mutual inductance between the *as* and *bs* stator windings is zero and the self-inductances L_{asas} and L_{bsbs} are equal and denoted L_{ss}. Utilizing L_{ss} and equations (3.4) through (3.7), the voltage equations (3.2) and (3.3) become

$$v_{as} = r_s i_{as} + p(L_{ss} i_{as} + \lambda'_m) \tag{3.8}$$

$$v_{bs} = r_s i_{bs} + p(L_{ss} i_{bs} + \lambda'_m) \tag{3.9}$$

or in matrix form

$$\mathbf{v}_{abs} = \mathbf{r}_s \mathbf{i}_{abs} + p(\mathbf{L}_s \mathbf{i}_{abs} + \lambda'_m) \tag{3.10}$$

The simulation of the permanent-magnet machine is made easier by transformation to a reference frame fixed in the rotor by use of the Park's transformation [Ref. 5, pp. 133-160]. In this transformation, the electrical frequency $\omega_e = \omega_r$ and therefore $\theta_e = \theta_r$. This will enable the use of a dc voltage input to control the machine speed rather than a sinusoidal voltage from a dc-to-ac inverter. The Park's transformation for a two-phase machine is

$$\begin{bmatrix} f^r_{qs} \\ f^r_{ds} \end{bmatrix} = \begin{bmatrix} \cos\theta_r & \sin\theta_r \\ \sin\theta_r & -\cos\theta_r \end{bmatrix} \begin{bmatrix} f_{as} \\ f_{bs} \end{bmatrix} \tag{3.11}$$

or

$$\mathbf{f}^r_{qds} = \mathbf{K}^r_s \mathbf{f}^r_{abs} \tag{3.12}$$

and

$$\mathbf{f}^r_{abs} = (\mathbf{K}^r_s)^{-1} \mathbf{f}^r_{qds} \tag{3.13}$$

where f represents either voltage, current, or flux linkage. The subscript s denotes stator variables and superscript r indicates transformation to the rotor reference frame. By substituting (3.13) into (3.10), the matrix voltage equation becomes

13

$$(\mathbf{K}_s^r)^{-1} \mathbf{v}_{qds}^r = r_s \mathbf{I} (\mathbf{K}_s^r)^{-1} \mathbf{i}_{qds}^r + p[L_{ss} \mathbf{I} (\mathbf{K}_s^r)^{-1} \mathbf{i}_{qds}^r + (\mathbf{K}_s^r)^{-1} \lambda_m^{'r}] \tag{3.14}$$

Pre-multiplying equation (3.14) by $\mathbf{K}_s^r$ and considering $\lambda_m^{'r}$ as a constant, the voltage equations may be written in matrix form as

$$\begin{bmatrix} v_{qs}^r \\ v_{ds}^r \end{bmatrix} = \begin{bmatrix} r_s + pL_{ss} & \omega_r L_{ss} \\ -\omega_r L_{ss} & r_s + pL_{ss} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix} + \begin{bmatrix} \omega_r \lambda_m^{'r} \\ 0 \end{bmatrix} \tag{3.15}$$

Since $\theta_e = \theta_r$

$$\begin{aligned} v_{qs}^r &= |v_{as}| \\ v_{ds}^r &= 0 \end{aligned} \tag{3.16}$$

and equation (3.15) becomes

$$\begin{bmatrix} v_{qs}^r \\ 0 \end{bmatrix} = \begin{bmatrix} r_s + pL_{ss} & \omega_r L_{ss} \\ -\omega_r L_{ss} & r_s + pL_{ss} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \end{bmatrix} + \begin{bmatrix} \omega_r \lambda_m^{'r} \\ 0 \end{bmatrix} \tag{3.17}$$

The electromagnetic torque is related to the rotor speed ω_r by

$$T_e = (J_r + J_{ml}) \frac{d\omega_r}{dt} + B\omega_r + T_{motload} \tag{3.18}$$

and to motor current i_{qs}^r by

$$T_e = \frac{P}{2} \lambda_m^{'r} i_{qs}^r \tag{3.19}$$

Combining equations (3.18) and (3.19), we obtain

$$\frac{P}{2} \lambda_m^{'r} i_{qs}^r = (J_r + J_{ml}) \frac{d\omega_r}{dt} + B\omega_r + T_{motload} \tag{3.20}$$

where J_r and J_{ml} represent the *inertia* of the rotor and the connected mechanical load respectively, B represents the *damping coefficient* due to the friction and windage losses, and $T_{motload}$ represents the load torque delivered to the motor.

2. The Motor as a System of State Variables

From equations (3.17) and (3.20), expressions for the relationship between the voltage equations and the electromagnetic torque can be written in terms of the state variables i_{qs}, i_{ds} and ω_r

$$\frac{di_{qs}^r}{dt} = -\frac{r_s}{L_{ss}} i_{qs}^r - \omega_r i_{ds}^r - \frac{\lambda_m^{'r}}{L_{ss}} \omega_r + \frac{1}{L_{ss}} v_{qs}^r \tag{3.21}$$

$$\frac{di_{ds}^r}{dt} = -\frac{r_s}{L_{ss}} i_{ds}^r + \omega_r i_{qs}^r \tag{3.22}$$

$$\frac{d\omega_r}{dt} = -\frac{B}{(J_r + J_{ml})} \omega_r + \left(\frac{P}{2}\right)^2 \frac{\lambda_m^{'r}}{(J_r + J_{ml})} i_{qs}^r - \left(\frac{P}{2}\right) \frac{1}{(J_r + J_{ml})} T_{motload} \tag{3.23}$$

By substituting the *electrical time constant* $\tau_a = L_{ss}/r_a$, the *back-emf constant* $k_e = \lambda_m^{'r}$, and the *torque constant* $k_t = \frac{3}{2}\left(\frac{P}{2}\right)\lambda_m^{'r}$ into the two-phase system described by the linear differential equations (3.21) through (3.23), the system becomes a three-phase model represented by

$$p\begin{bmatrix} i_{qs}^r \\ i_{ds}^r \\ \omega_r \end{bmatrix} = \begin{bmatrix} -\dfrac{1}{\tau_a} & 0 & -\dfrac{k_e}{r_a \tau_a} \\ 0 & -\dfrac{1}{\tau_a} & 0 \\ \dfrac{Pk_t}{2J_{tot}} & 0 & -\dfrac{B}{J_{tot}} \end{bmatrix} \begin{bmatrix} i_{qs}^r \\ i_{ds}^r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -\omega_r i_{ds}^r \\ \omega_r i_{qs}^r \\ 0 \end{bmatrix} + \begin{bmatrix} \dfrac{1}{r_a \tau_a} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\dfrac{P}{2J_{tot}} \end{bmatrix} \begin{bmatrix} v_{qs}^r \\ 0 \\ T_{motload} \end{bmatrix}$$

$$\tag{3.24}$$

The constants r_a, τ_a, k_e, and k_t are usually provided by the manufacturer's for a specific motor.

Two important observable values are the current drawn and the power delivered by the motor. The actual current drawn by the motor is related to the input by

15

$$V_{dc}I_{dc} = \frac{3}{\pi}(v_{qs}^r i_{qs}^r)$$ (3.25)

where V_{dc} and I_{dc} are the input voltage and current to the motor. Since $V_{dc} = v_{qs}^r$ then the current drawn by the motor is given by

$$I_{dc} = \frac{3}{\pi} i_{qs}^r$$ (3.26)

The actual power delivered by the motor is

$$P_{motor} = T_{motload}\left(\frac{2}{P}\right)\omega_r$$ (3.27)

The time-domain block diagram for the permanent-magnet dc machine represented by equation (3.24) and (3.26) is shown in Figure 3-6. The diagram represents the motor component of the vehicle system to be simulated.

16
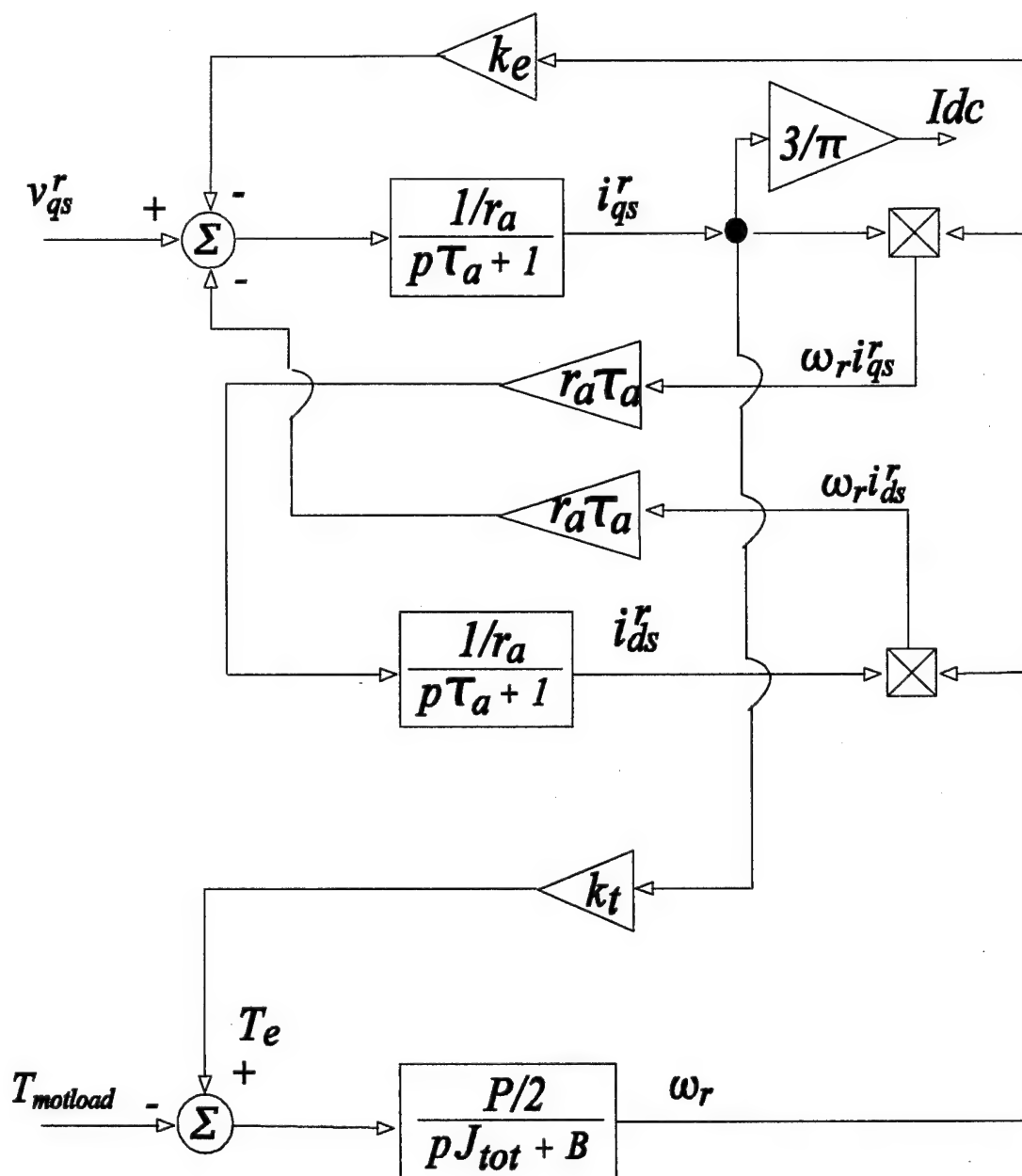
Figure 3-6. Time-domain block diagram implementation of the permanent-magnet brushless dc motor given in equation (3.24).

17

IV. MODELING THE VEHICLES MECHANICAL CHARACTERISTICS

A. THE FORCES ACTING ON THE VEHICLE

The forces that have been considered important in modeling an electric vehicle are described as the vehicles *performance variables* [Ref. 7, App. B, and Ref. 1, pp. 46-47]. These performance variables are comprised of forces acting on the vehicle and are used to calculate the power required to propel a vehicle at some steady velocity. The modeling equation which describes this power is

$$P_{motor} = \text{power required to drive the vehicle}$$
$$= \frac{P_R}{E_M E_E} + \frac{P_A}{E_A} \tag{4.1}$$

where, P_R = Power required at the rear drive wheel, P_A = Power required for accessories, E_A = Efficiency of the accessories, $E_M E_E$ = mechanical and electrical efficiency of the drive train and motor. By neglecting the accessories, we may express P_{motor} as

$$P_{motor} = \frac{P_R}{E_M E_E} = \frac{v(F_W + F_R + F_{HG} + F_A)}{E_M E_E} \tag{4.2}$$

where v = the vehicle velocity, F_W = wind force, F_R = rolling resistance force, F_{HG} = hill gravitational force, and F_A = acceleration force. These forces are describe the total drag on the vehicle, in Newton's and will be expanded upon. Equation (4.2) will be referred to as the *performance equation*.

1. The Wind Force (F_W)

This performance variable describes the force of friction on the vehicle body as it moves through the air. It is specified at zero wind velocity and is a function of the frontal surface area of the vehicle, as well as any protrusions on the vehicle, air passages in the body and any other components of the vehicle affected by movement over, around or through the vehicles shell or body. The ideal vehicle shape is a long cylinder with tapered ends. The equation that describes the wind force is:

$$F_W = \frac{1}{2} \rho_A v^2 C_d A \tag{4.3}$$

18

where ρ_A is the air density (1.2 kg/m³ was used in all calculations), v is the vehicle velocity (m/s), C_d is the coefficient of aerodynamic drag, and A is the vehicles frontal surface area (m²).

2. The Rolling Resistance Force (F_R)

The rolling resistance is primarily a function of the tire friction on the roadway. The coefficient of friction is directly related to the type of tire used. Narrow, high pressure tires will have a lower coefficient of friction than tires that are wider and lower pressure. Bicycle tires are normally used in solar powered vehicle applications. The equation which describes the rolling resistance force is:

$$F_R = M_V C_r g \tag{4.4}$$

where M_V = vehicle mass (kg), C_r = coefficient of rolling resistance, and g = gravitational constant (9.8 m/s²).

3. The Gravitational Force (F_{HG})

The hill gravitational force affects the total drag only when the vehicle is traversing a hill. Uphill equates to a positive grade and downhill a negative grade. The equation describing this force is:

$$F_{HG} = M_V g \, \sin \theta_{hill} \tag{4.5}$$

where θ = angle of the road measured from the horizontal component of the driving route.

4. The Acceleration Force (F_A)

The acceleration force is also known as the *inertial force* [Ref. 4, pp. 20-23]. This component of the total drag will have a positive sign for acceleration and negative sign for deceleration. The acceleration force equation is:

$$F_A = M_V a \tag{4.6}$$

where a = vehicle acceleration (m/s²). F_A is a dominant force in the process of *regenerative braking*. This is the process of recovering energy in the form of current supplied to the battery as discussed in Chapter III.

5. The Load Torque to the Motor

The forces expressed in the performance equation (4.2) relate directly to the load experienced by the motor. Realizing that a chain or belt drive translates these forces to

the motor tangent to the rotation of the drive wheel in contact with the road, the load torque T_{load} then becomes [Ref. 8, p. 183]

$$T_{load} = \frac{r \sum F_{vehicle}}{NE_M} \qquad (4.7)$$

where r is the radius of the drive wheel and N is the speed or gear ratio from the drive axle to the motor. Expanding the sum of vehicle forces represented in equation (4.7) gives

$$T_{load} = \frac{r(F_w + F_R + F_{HG} + F_A)}{NE_M} \qquad (4.8)$$

Figure 4-1 shows how the vehicle forces are translated to the motor.
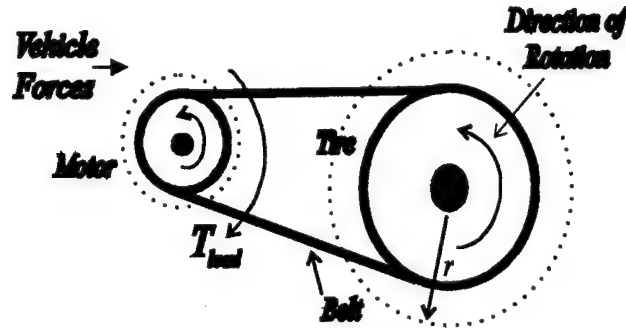
Figure 4-1. Interaction of vehicle force with motor for computing load torque T_{load}.

B. THE VEHICLE DRIVE TRAIN

The drive train acts as the coupling mechanism between the motor and the wheels. The most common method used in solar powered electric vehicles are chain or belt drives. The drive train model will be presented by assuming a belt drive configuration. The motors radial velocity in revolutions per minute is given as

$$RPM_{mot} = \left(\frac{2}{P}\right)\frac{\omega_r}{2\pi} \cdot 60 \qquad (4.9)$$

20

where ω_r is the rotor electrical frequency in radians per second, and P is the number of poles for the specific motor.

The RPM_{mot} is converted to the appropriate linear velocity by multiplication with the gear ratio and tire radius. The selection of the gear ratio depends on the things such as type of terrain over which the vehicle is traveling, the desired vehicle speed, the rated speed of the motor, etc. For example, over a level terrain a ratio of 1:5 may be utilized. If on the other hand, the terrain is uphill, an appropriate gear ratio would be 1:9. A higher gear ratio provides more power for climbing hills, but also greatly reduces the maximum speed of the vehicle. Utilizing equation (4.9), the gear ratio $G_R = \dfrac{1}{N}$, and the wheel radius r, the vehicle velocity is expressed by

$$v_{car}(kph) = RPM_{mot} \cdot \frac{60}{1000} \cdot \frac{1}{N} \cdot 2\pi r \qquad (4.10)$$

then converting to miles per hour

$$v_{car}(mph) = 0.625 \, v_{car}(kph) \qquad (4.11)$$

C. THE MECHANICAL COMPONENTS DESCRIBED AS A SYSTEM

The mechanical components comprising the vehicle forces which translate into the load torque T_{load} experienced by the motor, are presented in the block diagram shown in Figure 4-2. The input θ_{hill} is the angle of the incline or decline in radians. If the vehicle is traveling on level ground, then the input θ_{hill} would equal zero. The vehicles forward velocity component of the performance equation (4.2) is derived from the drive train. The velocity described in equation (4.11) is converted from kilometers per hour to meters per second.
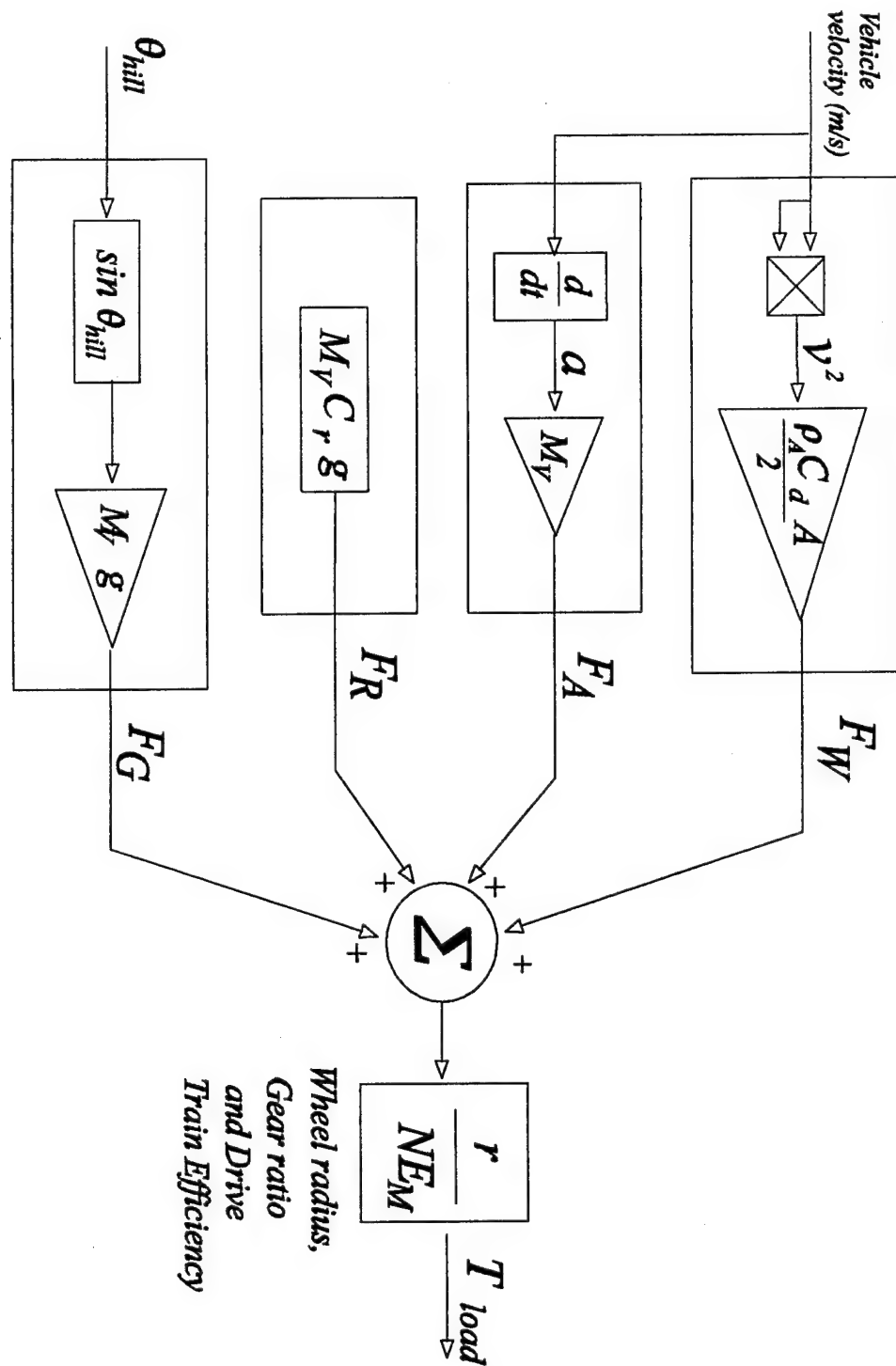
Figure 4-2. Block diagram of the load torque as a function of the forces acting of the vehicle.

The drive train equations (4.10) and (4.11) are presented in block diagram form in Figure 4-3.
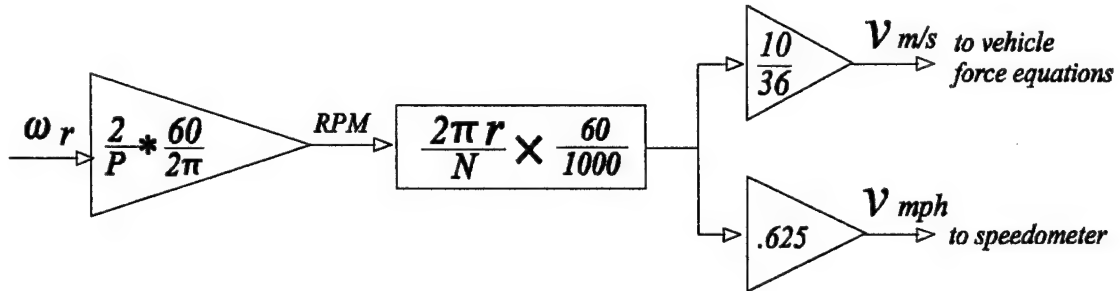
Figure 4-3. Block diagram of drive train conversion of motor rotational speed to vehicle forward velocity.

V. SIMULATION SOFTWARE AND MODEL DEVELOPMENT

A. THE SIMULATION SOFTWARE

Matlab [Ref. 2] was chosen as the simulation software for the vehicle system. The Matlab Toolbox - Simulink, is the primary building tool for designing and connecting the dynamic model components of the vehicle simulator. Simulink models both linear and non-linear system components in *block diagram "windows"*. These windows resemble the block diagrams presented in many control theory textbooks, which allows the user to easily construct and modify a system with a click of a mouse button. The model components presented in the previous chapters are assembled in Simulink block diagram submodels, then connected to present the overall vehicle system. Figure 5-1 shows the Simulink model for the entire solar-powered electric vehicle simulator system.

The user interface with the vehicle simulator is generated in MATLAB code (Appendix A). A user control window initiates the program and allows the user to input the vehicles characteristics by clicking the mouse on a "pushbutton" within the SPEV figure window. The required input information is displayed in the MATLAB control window. Once the vehicle data has been entered the simulation may be run and output data analyzed. The specific vehicle information required for input is discussed in Chapter VI. User instructions are provided in Appendix A.

B. THE ELECTRICAL COMPONENTS IN SIMULINK

In Chapter III, the individual components which comprise the electrical characteristics of the solar powered vehicle were presented as equations and block diagrams. These block diagrams will now be presented in Simulink form.

1. The Run Time and Solar Array Characteristics

The first components of the vehicle simulator are the *run time* and *solar array output*. Current vectors are developed from the three types of solar array configurations presented in Chapter III, which are the flat, curved, and tracking array configurations. An array type and expected peak current output are specified by the user, then the associated vector is read into the *Array* look-up table block. The time of day for simulation start is specified by the user and can be between 0900 and 1700. Figure 5-2 shows the simulink implementation of the time and array look-up table. The *variable cloud cover* block is used to simulate the loss of array output due to overcast conditions. The user can "double click" on the block, opening it to

24
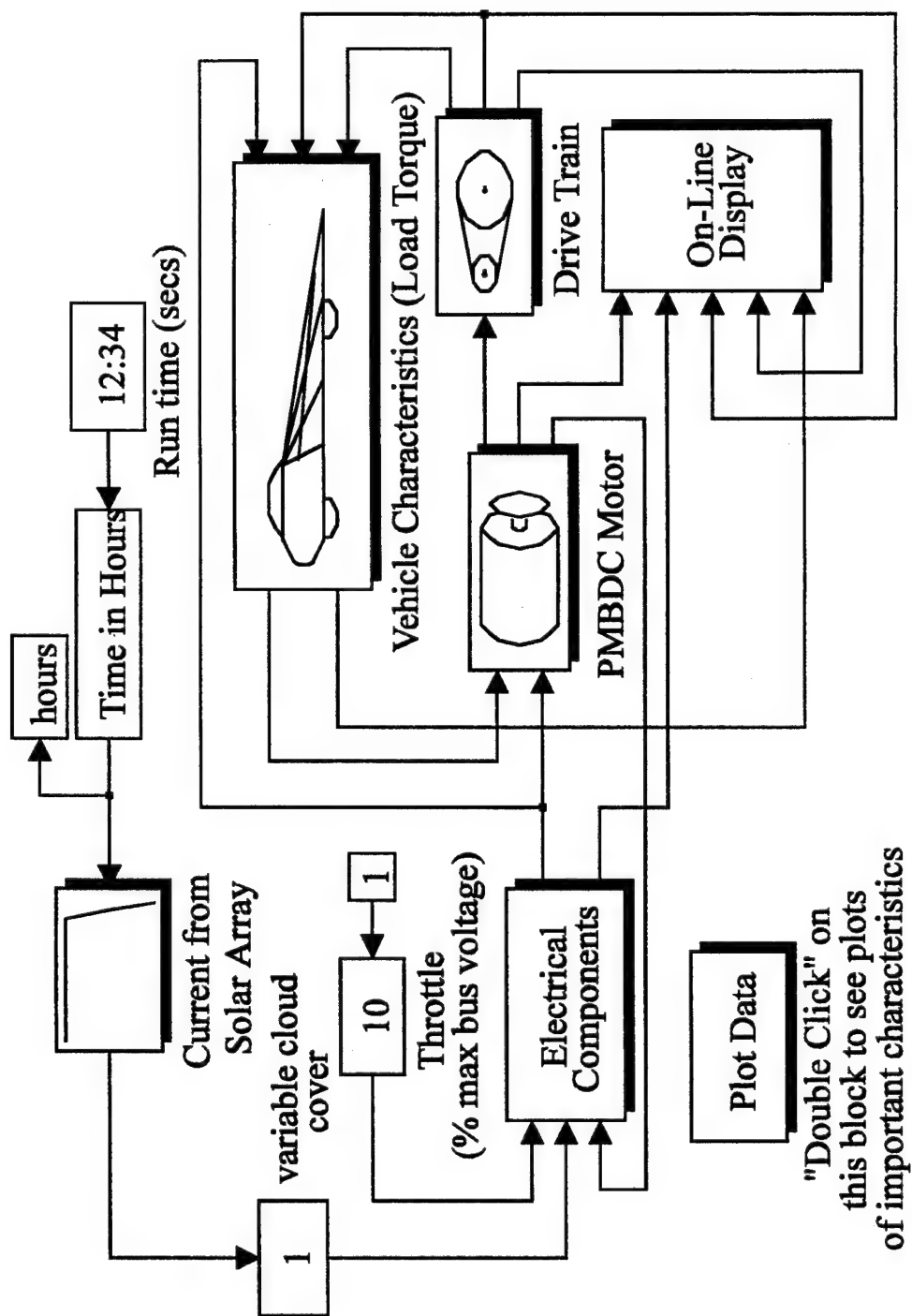
Figure 5-1. Solar-Powered Electric Vehicle simulator in Simulink.

25

a slider gain. Clicking on the down or up arrow changes the gain coefficient which produces changes in the array current output fed to the *Electrical Components* block, thereby simulating current degradation due to cloud cover.
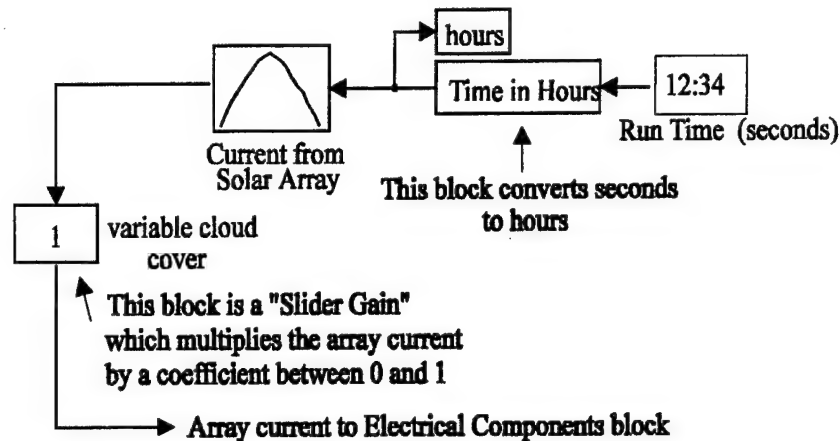
Figure 5-2. Simulink implementation of the Solar Array.

2. The Electrical Components Block

The *Electrical Components* block shown in Figure 5-3 takes input from the vehicle throttle, which is a slider gain, current fed from the array as previously discussed, and current being drawn by the drive motor. The output is controlled dc voltage to the motor from the battery, and a multiplexed array of outputs to be monitored in the *On-Line Display*. These outputs include:

- state-of-charge
- total amps-hours used
- battery voltage
- motor current drawn
- array current delivered

The *Battery* block uses a look-up table to determine voltage output based on state-of-charge as presented in equation (3.1). The user specifies the type of battery to be used, either Lead-Acid or Silver-Zinc. The discharge curve for the chosen battery will be displayed in the *Battery* block. Also seen in Figure 5-3 are the contents of the *Electrical Components* block, which will be expanded on.

26
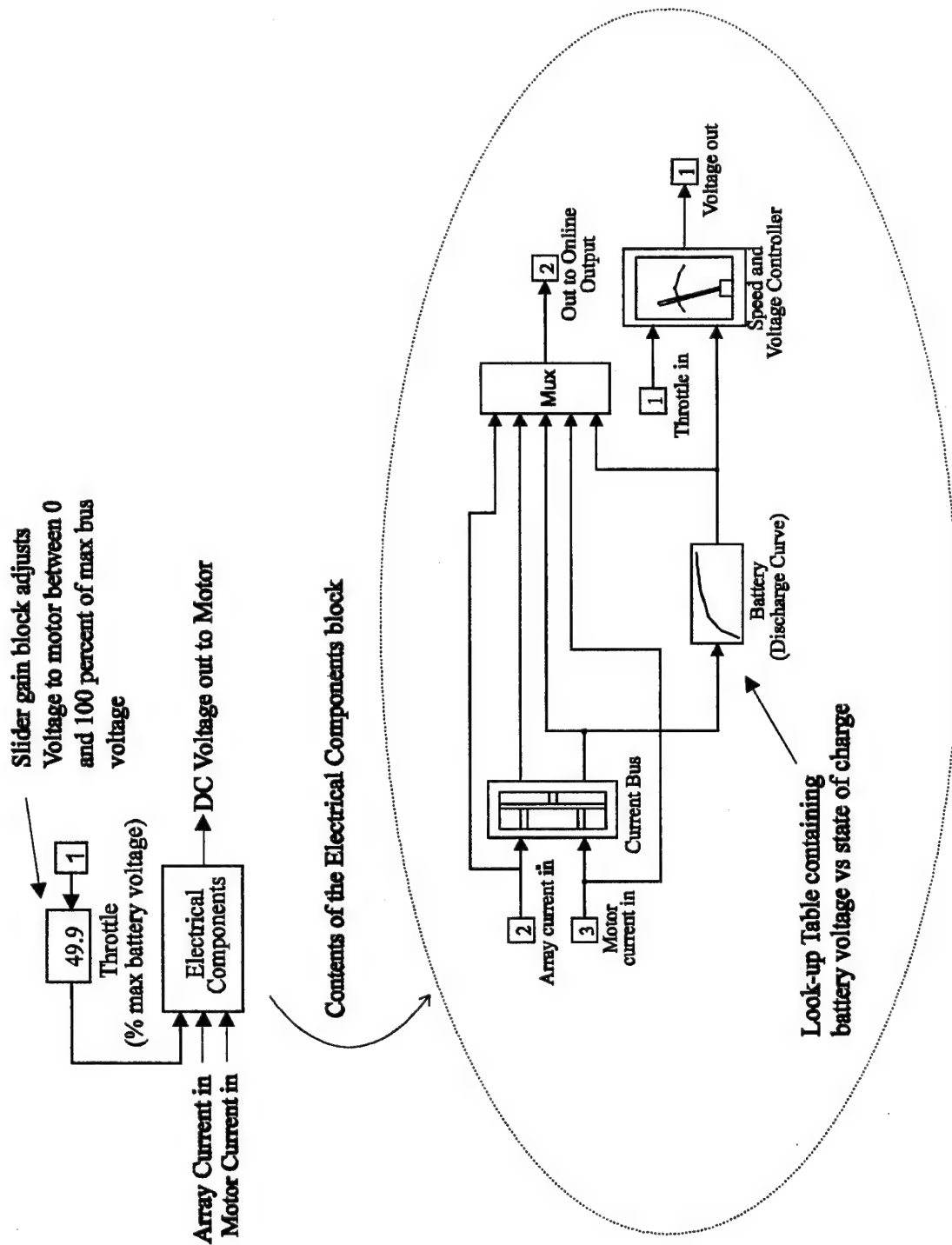
Figure 5-3. Simulink representation of the main electrical components of the vehicle simulator.

a. The Current Bus Block

The *Current Bus* block takes input from the solar array and the current being drawn by the motor, then uses these to compute the battery state-of charge *(SOC)* and the amp-hours (Ahrs) used. Figure 5-4 shows the expanded view of the *Current Bus* block. As developed in chapter 3, the sum of the array current and motor current are integrated then, divided by 3600 seconds to provide the amp-hour usage output. Since the total amp-hour usage cannot be less than zero, a switch is used to "turn off" the amp-hour output at a value of zero. The *State-of-Charge* function block implements equation (3.1) by using the amp-hour input to compute the battery state-of-charge based on maximum battery capacity and initial state-of-charge.

b. Speed and Voltage Controller Block

The inputs to the *Speed and Voltage Controller* block are throttle input and battery voltage. The output is controlled dc voltage to the motor to regulate speed. Figure 5-5 shows the expanded Simulink block. The throttle input, from a slider gain is sent to the *voltage ratio* function block which outputs a coefficient from 0 to 1. The throttle coefficient is rate limited before being multiplied by the battery voltage. The maximum voltage to the motor is set by a saturation block at 100 volts. The rate limited or "throttled" voltage is then sent to the motor.

3. The Drive Motor Block

The inputs to the *Drive Motor* block are controlled voltage from the *Electrical Components* block and vehicle load (T_{load}). The vehicle load will be discussed in following sections. The outputs are RPM to the drive train and a multiplexed array of RPM, motor load, efficiency and current drawn by the motor to the *On-Line Display*. The *Drive Motor* block is presented in Figure 5-6 and its contents are shown in Figure 5-7.

a. The System Efficiency Block

The *System Efficiency* block consists of a two dimensional look-up table with inputs of motor rpm and vehicle load, and switches for minimum and maximum efficiency. The efficiency limits are set by the minimum and maximum efficiency for the specific motor being used. The *System Efficiency* block is presented in Figure 5-8. The actual vehicle load is divided by the efficiency output in the *Load/Eff* function block. The output of this function block relates to the $T_{motload}$ introduced in equation (3.18), and used in equation (3.27) to determine motor power out. This is the vehicle load torque seen by the motor.

28
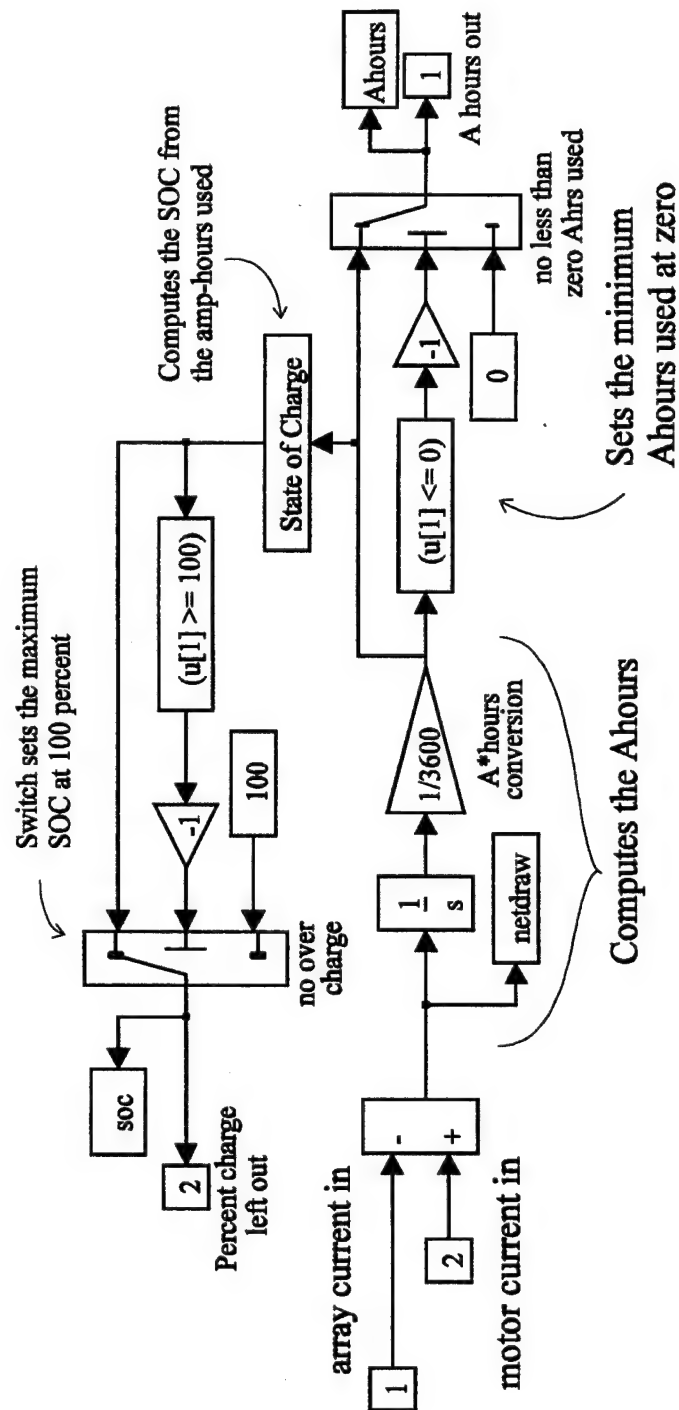
Figure 5-4. Simulink representation of Ahrs and SOC output from current input.
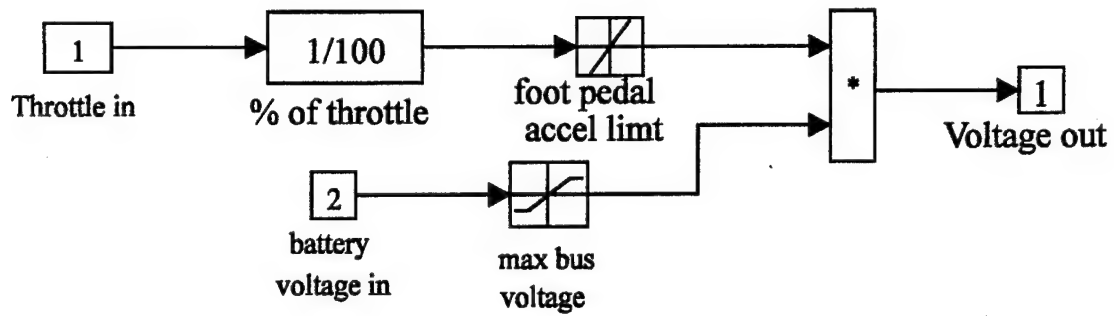
Figure 5-5. Simulink representation of Speed and Voltage control.
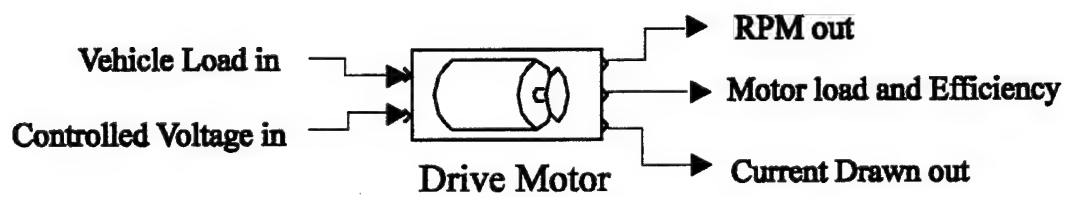
Figure 5-6. Simulink diagram of the Drive Motor block.
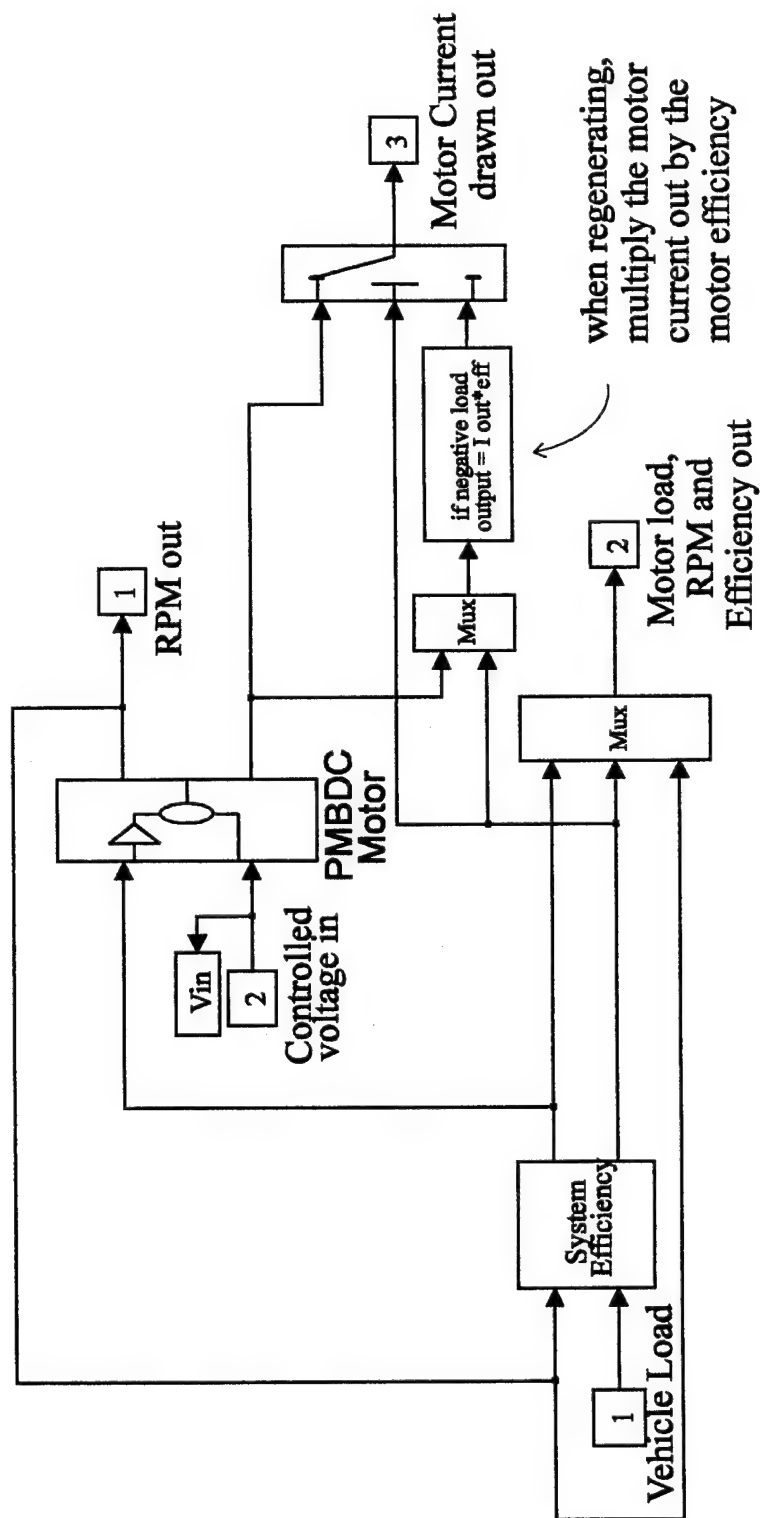
Figure 5-7. Contents of the Drive Motor block.

31

Figure 5-8. Simulink representation of System Efficiency block.

b. The PMBDC Block

The *PMBDC* block represents the permanent-magnet brushless dc motor used in the vehicle and implements equation (3.24) in Simulink form. The Simulink diagram of the motor is presented in Figure 5-9. The magnitude of the dc voltage through input 2 directly controls the desired rpm of the motor. The actual motor speed is of course affected by the load torque introduced by the mechanical components of the vehicle and the inertial load (J_{tot}). The speed in rpm is output to the *Drive Train* block of the simulator. The motor current drawn, rpm and load torque are sent to the On-Line display to be monitored during simulation.

C. THE MECHANICAL COMPONENTS IN SIMULINK

In Chapter IV, the individual components which incorporate the mechanical characteristics of the solar powered vehicle were presented as equations and submodels. The Simulink form of these submodels will now be presented and their operation explained.

1. The Vehicle Drive Train Block

The *Vehicle Drive Train* block uses the rpm input to convert rotational velocity of the motor to the linear velocity of the vehicle. The Simulink system through which this is accomplished is presented in Figure 5-10. The rpm delivered from the *Drive Motor* block through input 1 is divided by a specified gear ratio and mechanical efficiency, then multiplied by the circumference of the drive wheel. This provides a linear velocity in meters per minute. The linear velocity is converted to meters per second, for use in the *Vehicle Characteristics* block in determination of forces, and miles per hour. The miles per hour variable is integrated to provide the total miles traveled. The speed, in miles per hour, and total distance traveled are sent to the *On-Line Display* and monitored as the speedometer and odometer. The total miles traveled is also sent to the *Vehicle Characteristics* block for force computations.

2. The Vehicle Characteristics (Load Torque) Block

The *Vehicle Characteristics (Load Torque)* block is simply the implementation of the performance equation (4.2) and the associated torque equations detailed in section 4.5. The inputs to the *Vehicle Characteristics* block are the velocity in meters per second, total miles driven, and the voltage delivered to the motor. Each of the forces developed by the motion of the vehicle are summed and sent to the switch block. At this switch block the total force is either zero, if no voltage and no velocity are present, or the sum of all forces
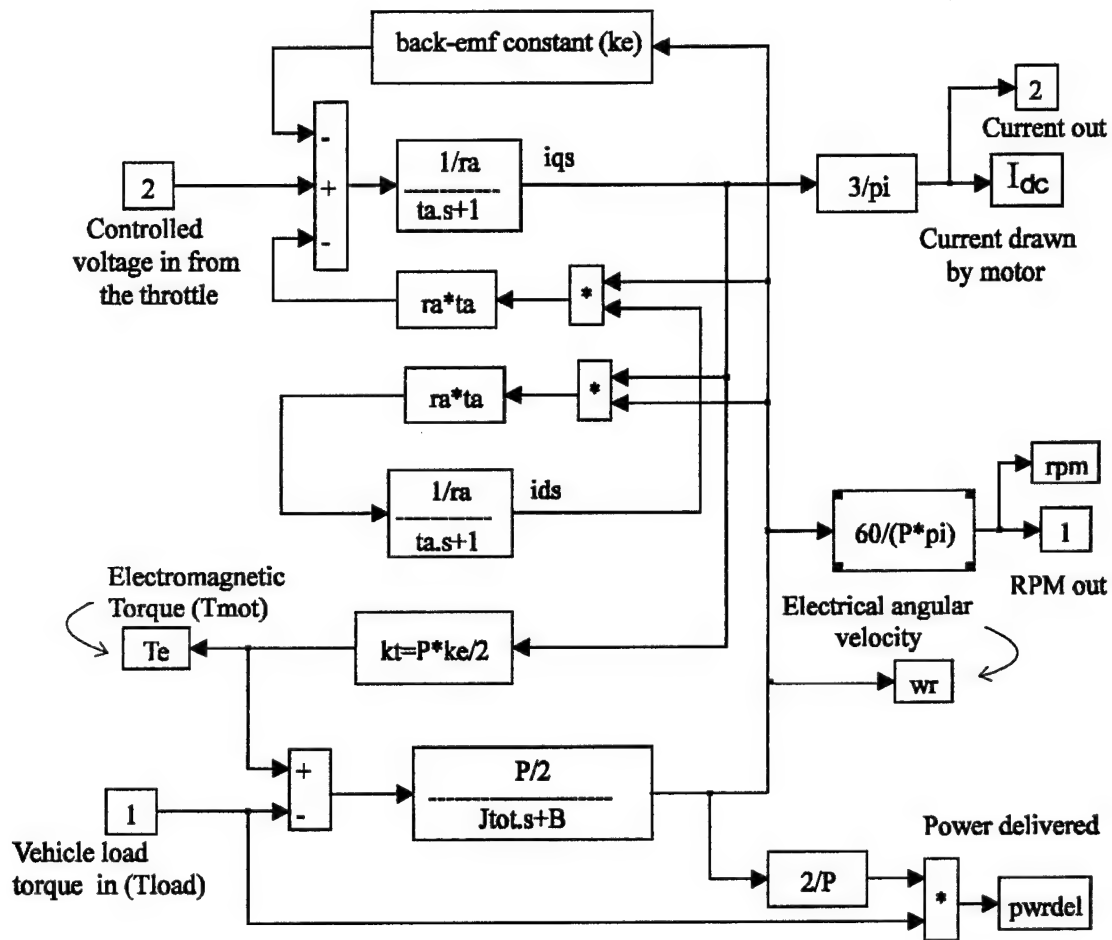
Figure 5-9. Simulink representation of a Permanent-Magnet Brushless DC Motor (PMBDC).

34
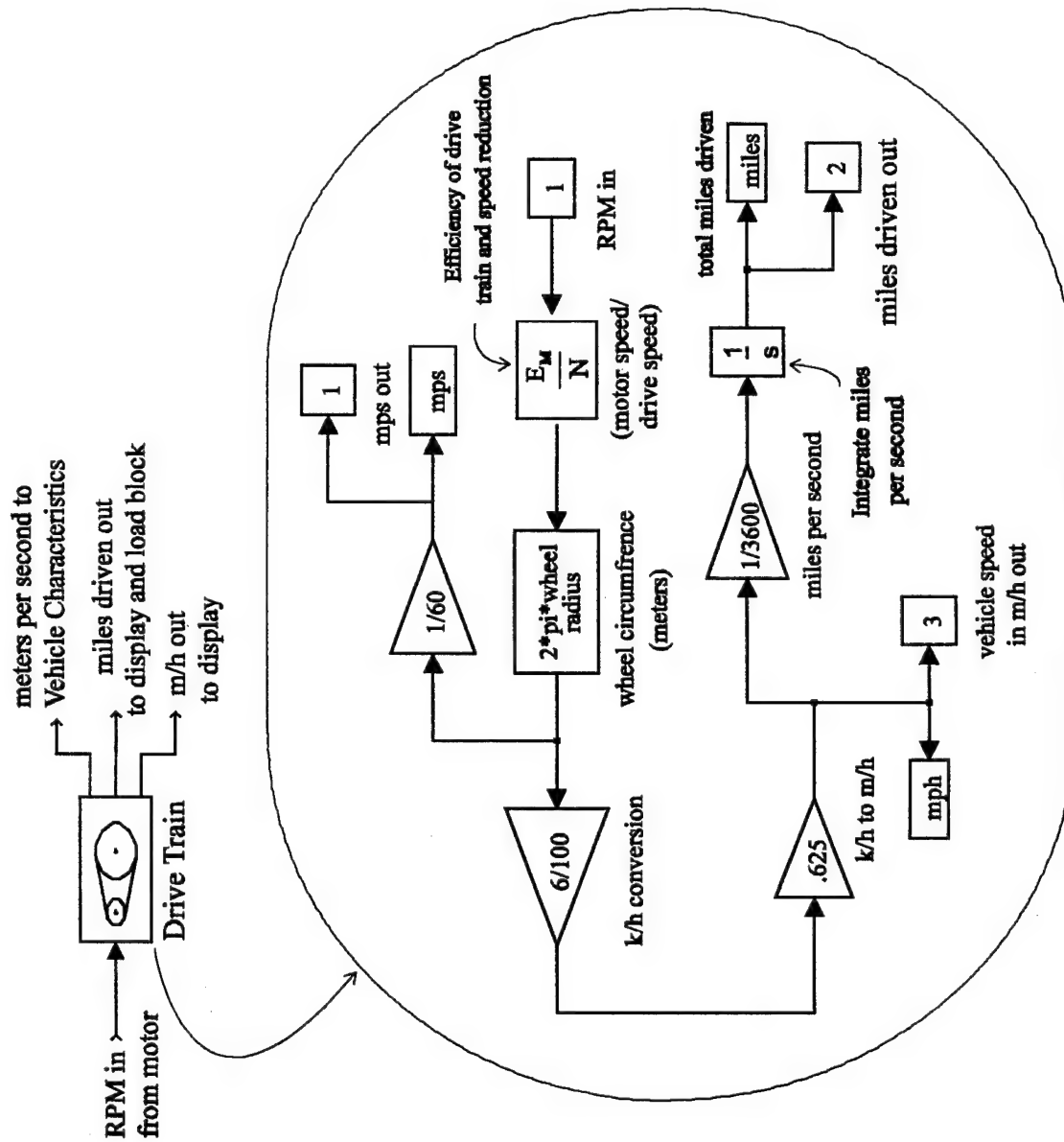
Figure 5-10. Simulink representation of the vehicle drive train.

35

present. Then, since the total force is actually applied at the point where the wheel contacts the road, the force is multiplied by the wheel radius and divided by the gear ratio, providing the vehicle load torque (T_{load}) seen by the motor. The complete contents of the *Vehicle Characteristics* block are shown in Figure 5-11. Each of the force components within the vehicle characteristics block will be presented and expanded upon.

a. The Rolling Resistance Block

The contents of the *Rolling Resistance* block are shown in Figure 5-12. The input to the block is the linear velocity of the vehicle and the hill force. The switches shown have three purposes: 1) if the vehicle velocity is zero, then the rolling resistance force is zero, 2) if the vehicle velocity is greater than zero, the rolling resistance force is applied against the direction of motion, and 3) if the hill force is greater than the rolling resistance force, and the vehicle is moving backwards, the rolling resistance force is applied in the opposite direction of motion. The rolling resistance force is a constant and not a function of the vehicle velocity.

b. The Hill Gravitational Block

The *Hill Gravitational* block computes the force on the vehicle as a function of hill angle. If the vehicle is traveling up-hill, the force is positive and increases the total load on the motor. If the vehicle travel is down-hill, the force is negative and consequently decreases the total load on the motor. If the vehicle is driving over level terrain, then the hill force is zero. This component plays a significant role in the regenerative action of the motor. If the hill angle is large enough to overcome the other forces applied to the motor, then the load of the vehicle will "drive" the motor, which in turn will "generate" current for battery charging. Computation of the hill force in Simulink is shown in Figure 5-13.

c. The Acceleration Force Block

The input to the *Acceleration Force* block is the linear velocity of the vehicle. The velocity is differentiated to obtain the vehicle acceleration used in determining the accelerating or decelerating force. As with the hill force, the acceleration force also plays a significant role in the regenerative action of the motor. If the voltage applied to the motor is decreased, the vehicle will begin to slow down. This action causes a large deceleration, or negative force, to be applied to the motor, which in turn creates a significant regeneration of current to the battery for charging.
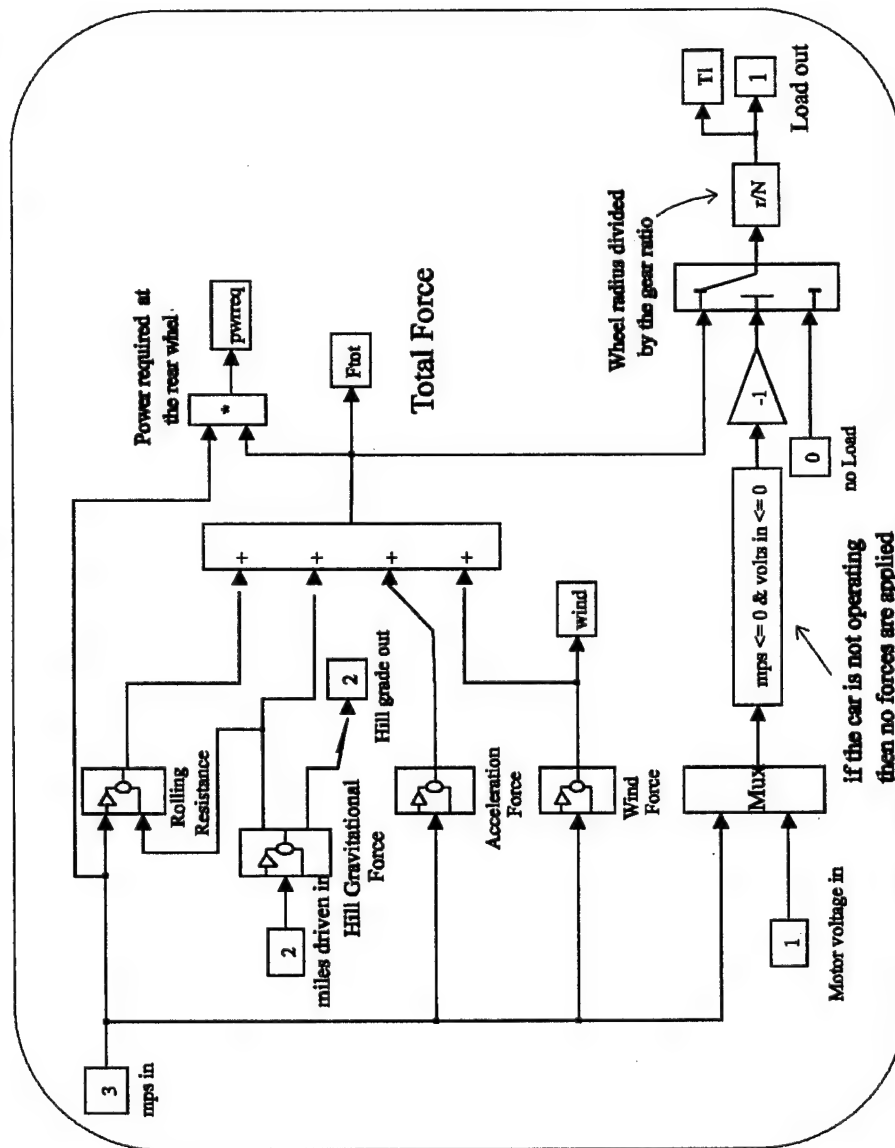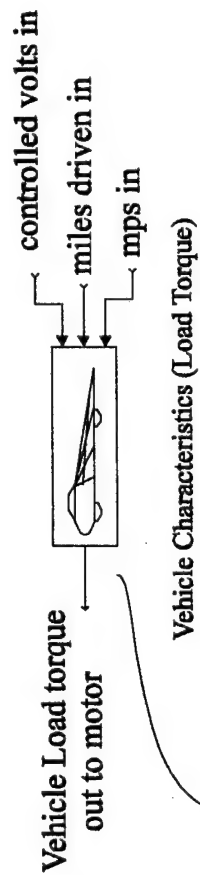
Figure 5-11. The Vehicle Characteristics block and contents.

37
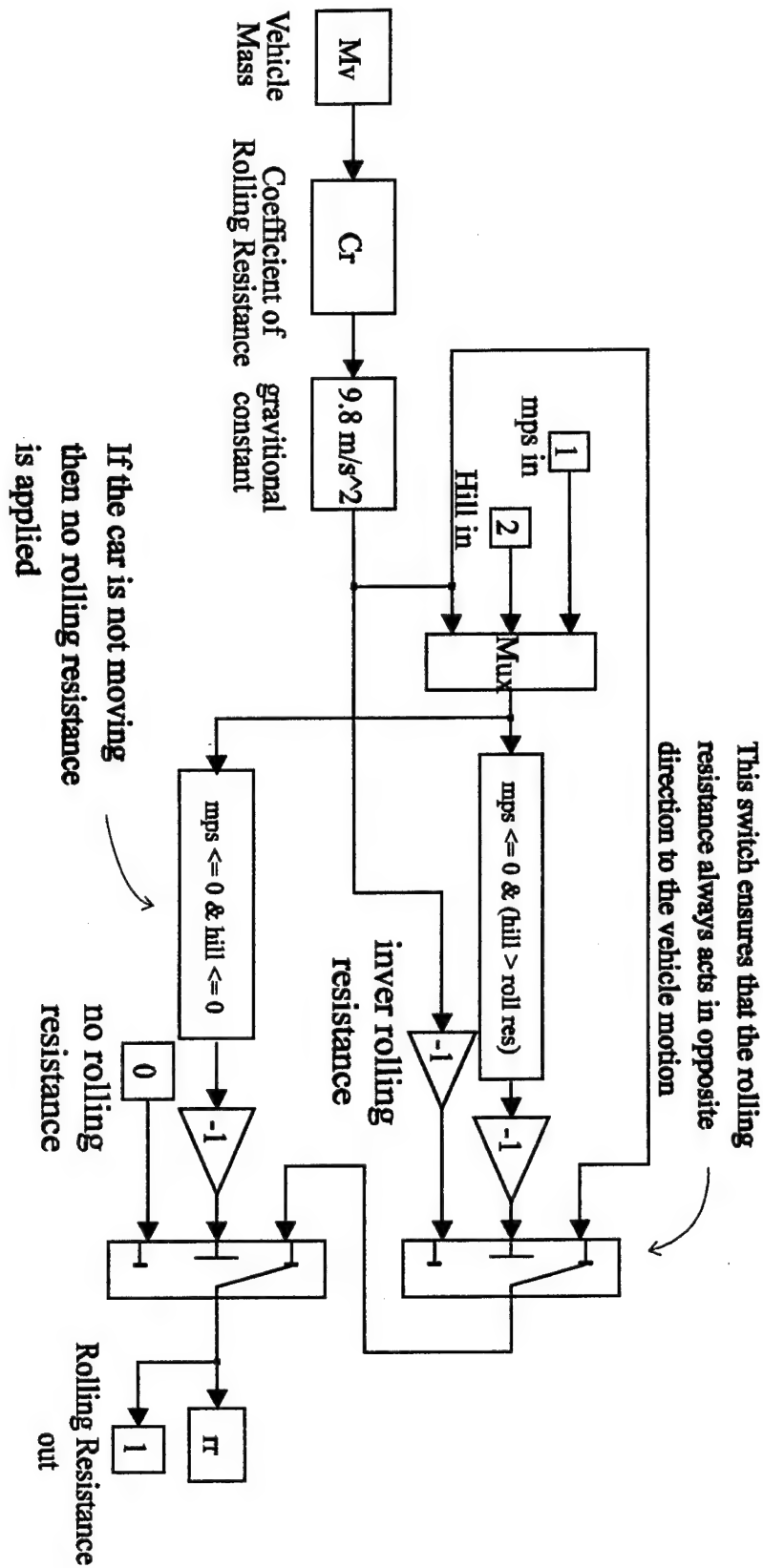
Figure 5-12. Simulink representation of the Rolling Resistance force.

38
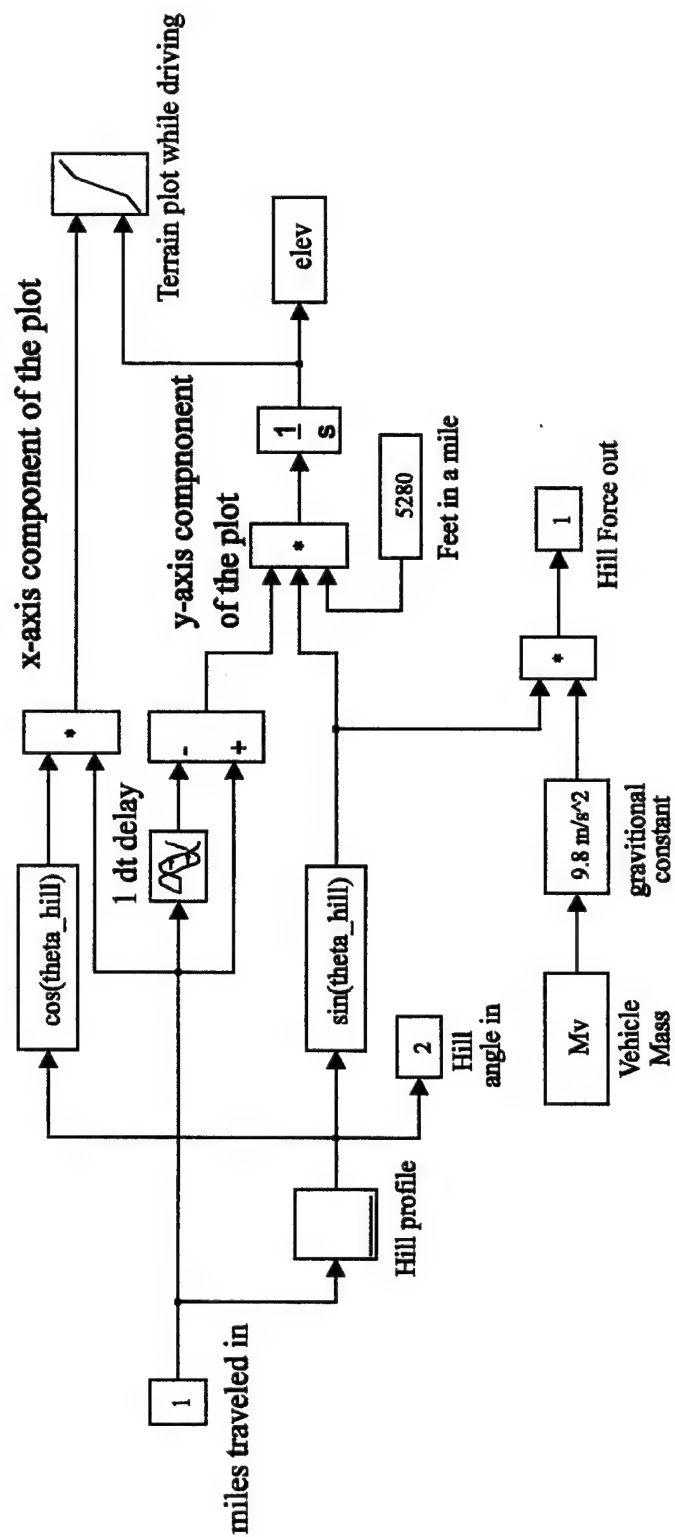
Figure 5-13. Simulink representation of the Hill Force computation.

This force is only applied during transient operation and has no steady-state component. Figure 5-14 shows the Simulink representation of the acceleration force computation.
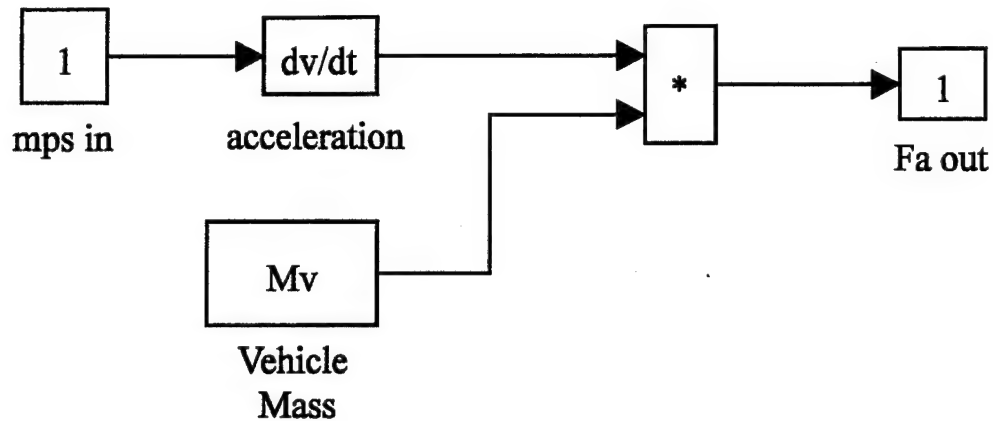
Figure 5-14. Simulink representation of the Acceleration Force computation.

d. The Wind Force Block

Figure 5-15 shows the Simulink representation of the wind force. The input to the *Wind Force* block is the linear velocity of the vehicle.
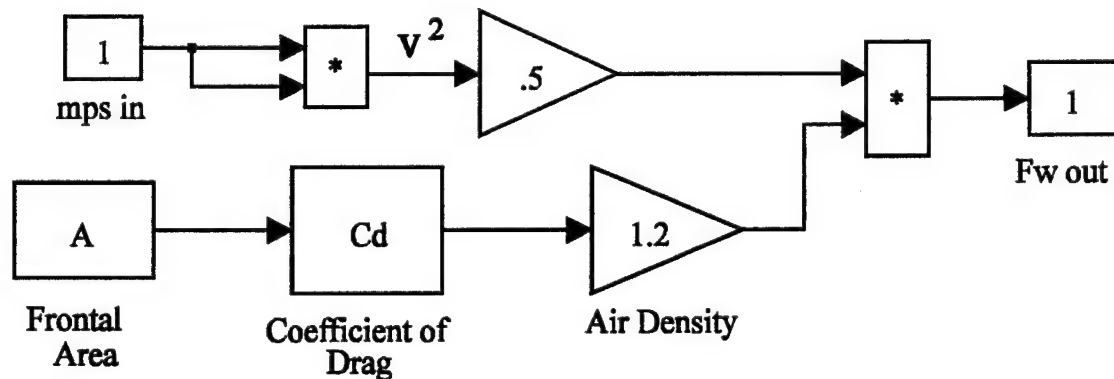
Figure 5-15. Simulink representation of the Wind Resistance computation.

This force is directly proportional to the square of the vehicles linear motion. The faster the vehicle goes, the greater the force.

D. THE VEHICLE ON-LINE DISPLAY IN SIMULINK

The *On-Line Display* block takes inputs from various components of the simulation program. These inputs are all feed into a simulink multiplex block and then sent to a Matlab s-function called *spevan2.m*. The function *spevan2.m* is included in Appendix B with the other Matlab functions used in the vehicle simulation program. The *On-Line Display* block and its components is presented in Figure 5-16. All inputs are clearly labeled.

Once the simulation routine has been initiated by the user, the *On-Line Display* window will appear at the bottom of the monitor. The outputs to the screen are clearly identified. Upon completion of a simulation run a "double-click" on the *Plots* block will provide a profile of important data and average values of certain characteristics. These average value components will be detailed in the following chapter.
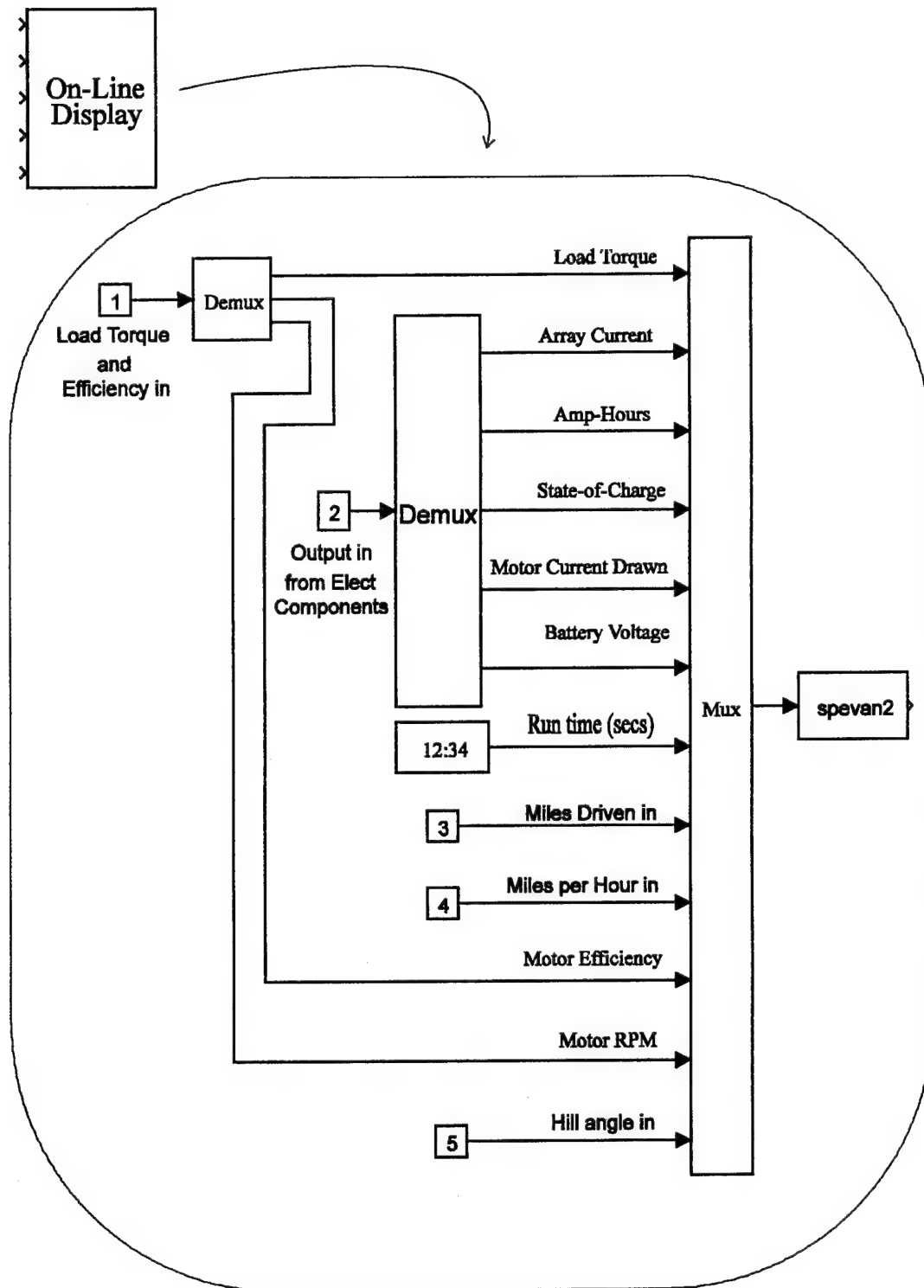
Figure 5-16. On-Line Display block and contents in Simulink.

VI. TESTING AND EVALUATION OF THE SYSTEM MODEL

A. OVERVIEW OF THE TESTING APPROACH

The primary question to ask in testing the vehicle simulation program is, "Does the power out of the motor equal the power required to overcome the forces acting on the vehicle for a specific speed?" This chapter will be concerned with answering this question over three types of terrain profiles - flat, single hill, and variable elevation.

1. Specification of the Initial Simulation Parameters

The vehicle electrical and mechanical simulation parameters where compiled from actual solar powered vehicles built by teams entered in the 1990 and 1993 World Solar Challenge.

The mechanical characteristics were complied from a vehicle built and raced by the CALSOL solar powered vehicle race team from the University of California at Berkeley. Table 6-1 shows the performance characteristics used for the vehicle simulator testing. These characteristics also serve as the default values for initiation of all simulations.

Coefficient of Drag (C_D)	.16
Frontal Area (A)	1.2 m^2
Vehicle Mass (M_V)	295.8 kg
Coefficient of Rolling Resistance (C_R)	.0105
Rear Wheel Radius (r)	.26 m
Gear Ratio (1:N)	1:5
Drive Train Efficiency	98 %

Table 6-1. Performance characteristics used in the vehicle simulation.

A permanent-magnet brushless dc motor built by Unique Mobility, Inc., is used in this vehicle simulation. The characteristics of the motor used in the simulation are presented in Table 6-2. Figure 6-1 shows the efficiency map provided for the motor. This efficiency map is loaded into the program upon initialization of the motor parameters for use during simulation. As with the vehicle characteristics, the motor parameters used for simulation testing also serve as the default values for simulator initialization.

Unique Mobility DR127s/CR10-100	
Voltage In (V_{dc})	50 volts
Rated Power	11.3Hp/8.4KW
Max No-Load Speed	5000 rpm
Number of Poles (P)	24
Damping Coefficient (B)	0
Rotor Inertia (J)	.014 Kgm^2
Electrical Time-Constant (τ_a)	6.944e-4
Winding Resistance (r_a)	3.6e-2 ohms
Back-EMF Constant (k_e)	20 V/Krpm
Torque Constant (k_t)	.191 Nm/A

Table 6-2. PMBDC Motor parameters used for simulation testing.
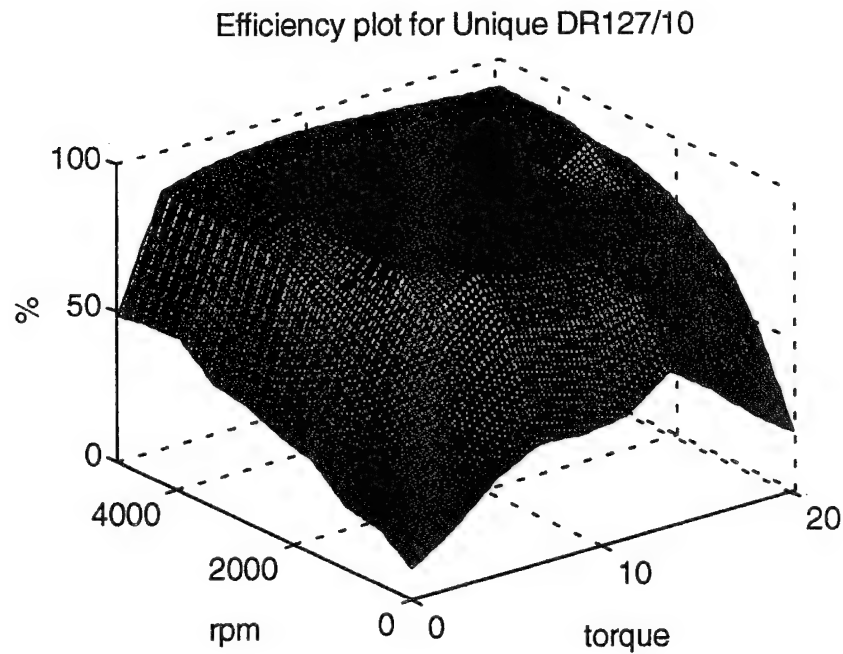
Figure 6-1. Motor efficiency plot.

The battery chosen for simulator testing is a Lead-Acid type, with a capacity of 35 Ahrs and initial state-of-charge of 100%. A tracking solar array will be used, with a peak current output of 11 amps. This battery and array configuration are the default components used in simulation initialization.

B. MODEL SIMULATION

The three types of terrain considered during testing are flat, single hill, and variable terrain which contains numerous positive and negative grades. The following outputs will be presented:

- driving terrain profile
- vehicle speed
- sum of the forces on the vehicle
- current drawn by the motor
- torque seen by the motor
- power delivered by the motor
- power required at the rear drive wheel

1. Simulation Over Flat Terrain

The first driving profile for simulation is over flat terrain as shown in Figure 6-2.
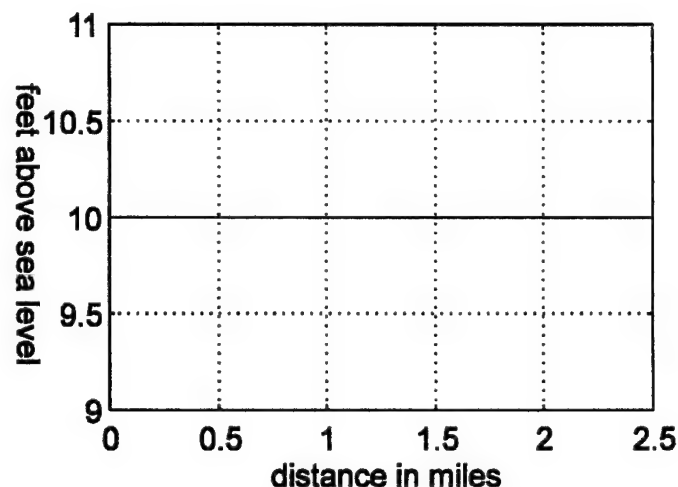
Figure 6-2. Plot of flat terrain driving profile.

The simulator system was initialized with the parameters specified in the previous section. Cloud cover was neglected, so peak current from the solar array was 11 amps.

45

The first outputs examined are the vehicle speed and total force on the vehicle. The theoretical speed is calculated from a quadratic equation based on the relationship between the energy delivered by the motor and the energy required to overcome the vehicle forces. This is expressed as

$$A + B(rpm_{mot}^2) = C - D(rpm_{mot})$$

(6.1)

$$A = \frac{M_V g C_R}{N E_M}, \quad B = \frac{\rho_A C_D A 2\pi^2 r^2}{60^2 N^3 E_M}, \quad C = \frac{V_{dc} k_t}{r_a}, \quad D = \frac{k_e k_t}{r_a}$$

With data provided in Tables 6-1 and 6-2, we get

$$rpm_{mot}^2 + \frac{0.1055}{0.000000607} rpm_{mot} + \frac{(6.2 - 263.9)}{0.000000697} = 0$$

(6.2)

Then using the quadratic equation

$$rpm_{mot} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$= -153770 \quad or \quad 2404.5$$

(6.3)

The motor rpm in (6.3) equates to a vehicle speed of v_{car} = 29.4 mph or 13.1 m/s. This value is within 95% of the simulated value presented in Figure 6-3. The slight variation is most likely caused from not considering the motor efficiency in the calculation.

Using a speed of 13.1 m/s, the theoretical total steady state force acting on the vehicle is derived from equation (4.2) as

$$F_{vehicle} = F_w + F_R$$

$$= M_V g C_R + \frac{1}{2} \rho_A C_D A v^2$$

$$= (295.8)(9.8)(0.0105) + \frac{1}{2}(1.2)(0.16)(1.2)(13.1^2)$$

$$= 50.2(Newtons)$$

(6.4)

miles per hour

Final value: 28.01

Total force on the vehicle (N)

Final value: 48.3
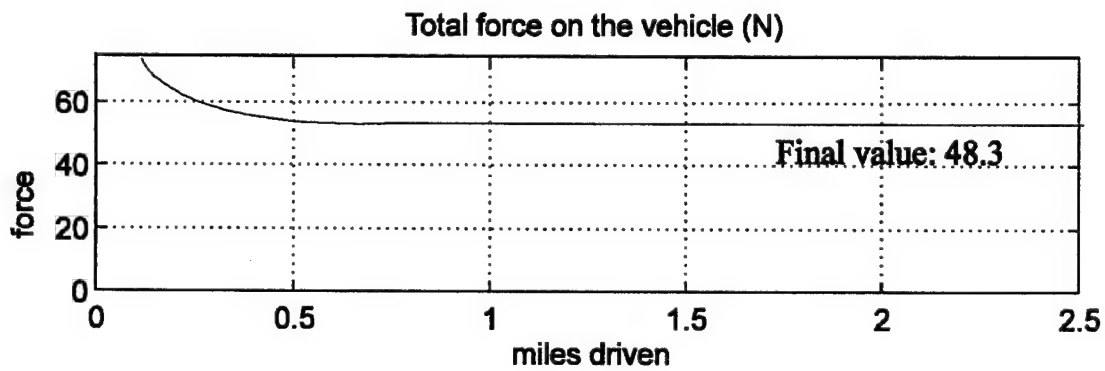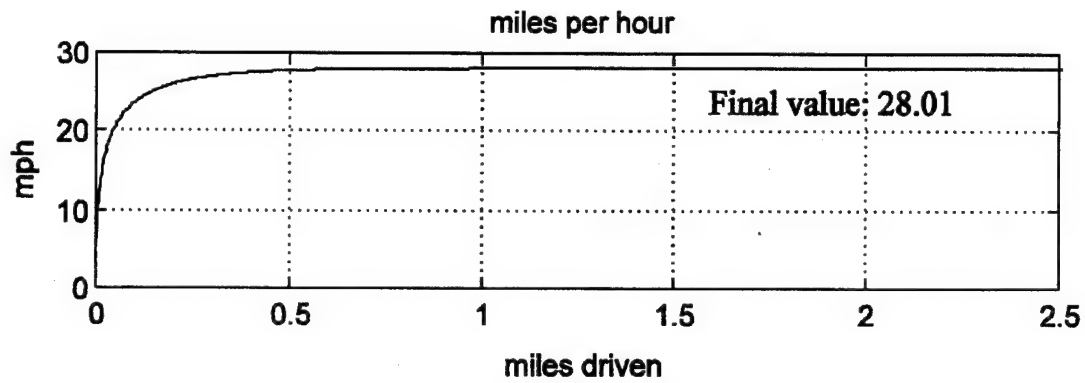
Figure 6-3. Plots of the steady state speed and total force acting on the vehicle for flat terrain driving

47

The acceleration force (F_A) and hill gravitational force (F_{HG}) are zero for flat steady state driving. The force given by (6.4) is within 96% of the simulated steady-state value of the total force displayed in Figure 6-3.

Also observed during the simulation are the current drawn by the motor and the torque delivered to the motor. Using equation (4.8), the theoretical force provided in (6.4), and data given in Table 6-1, the theoretical load torque delivered to the motor is

$$
\begin{aligned}
T_{load} &= \frac{r(F_w + F_R)}{NE_M} \\
&= \frac{(0.26)(50.2)}{(5)(0.98)} \\
&= 2.6(Newton - meters)
\end{aligned}
\tag{6.5}
$$

Utilizing the motor rpm from (6.3) and the load torque from (6.5), the motor efficiency may be determined from the efficiency plot in Figure 6-1. The efficiency is approximately 75%. Dividing (6.5) by the efficiency, the load torque delivered to the motor is

$$
T_{motload} = \frac{2.6}{0.75} = 3.47(Newton - meters)
\tag{6.6}
$$

Then utilizing (6.6) and equation (3.24) the theoretical current drawn by the motor is

$$
Idc = \left(\frac{3}{\pi}\right)i_{qs}^r = \left(\frac{3}{\pi}\right)\left(\frac{T_{motload}}{k_t}\right) = \left(\frac{3}{\pi}\right)\left(\frac{3.47}{0.191}\right) = 17.34(Amps)
\tag{6.7}
$$

As seen in Figure 6-4, the theoretical load torque and motor current drawn correspond closely to the steady-state output from the simulated model. Both values are within 97% of expected results.

The most important comparison to make is the power required to drive the vehicle versus the power delivered by the motor. The theoretical power required is computed from the performance equation (4.2). Utilizing the vehicle velocity from (6.3) and the total force on the vehicle given in (6.4), the power required at the rear wheels is

$$
P_R = v(F_W + F_R) = (13.1)(50.2) = 657.62(Watts)
\tag{6.8}
$$

Dividing (6.8) by the motor efficiency and the drive train efficiency, the power
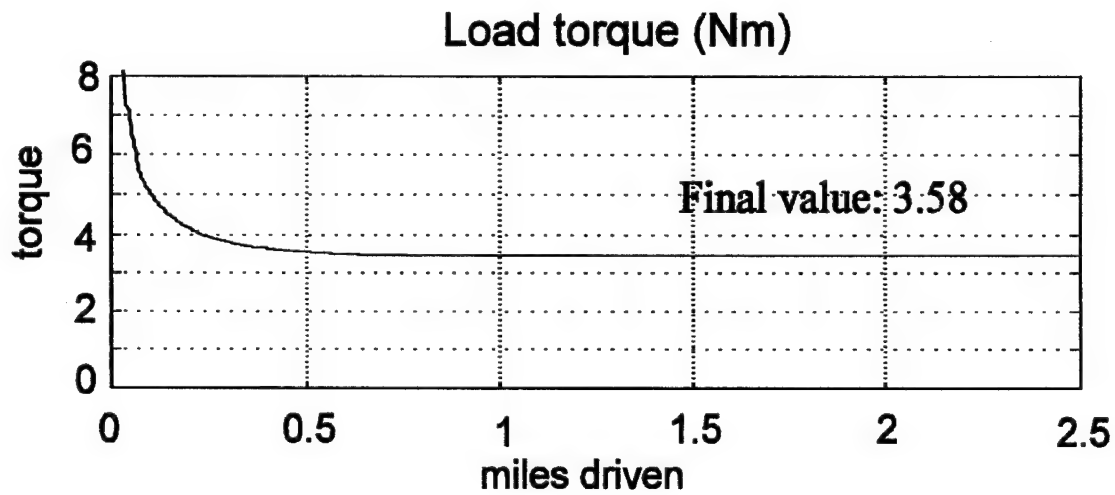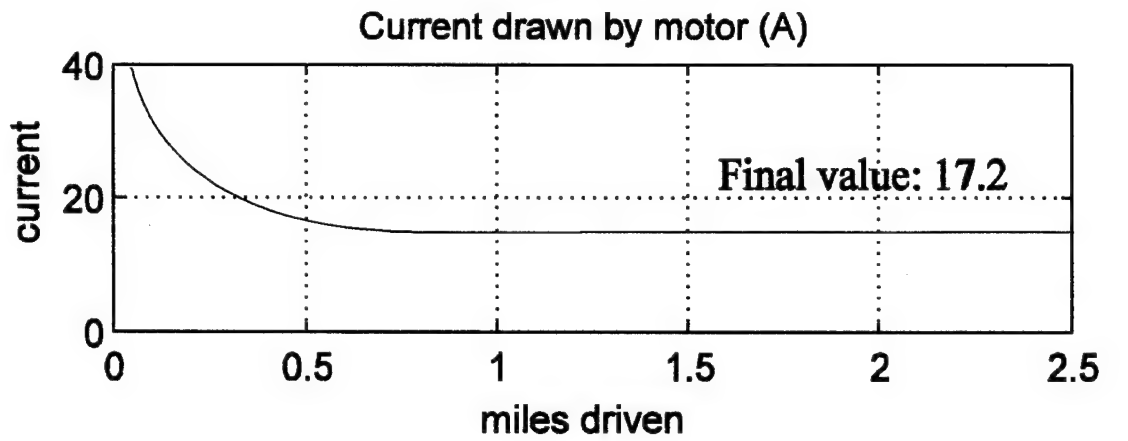
Figure 6-4. Plot of the motor current and the load torque delivered to the motor for flat terrain driving.

required from the motor is

$$P_{motor} = \frac{P_R}{E_E E_M} = \frac{657.62}{(0.75)(0.98)} = 894.72\,W \qquad (6.9)$$

The output plots of the power delivered by the motor and the power required at the rear wheels with consideration for system efficiencies are presented in Figure 6-5. The simulated steady-state values are within *94%* of the computed power in (6.9).

2. Simulation Driving Over a Single Hill Profile

The next simulation was up a single hill. Again, the array current was a constant 11 amps. The terrain profile for this simulation is shown in Figure 6-6. Using (6.1), with

$$A = \frac{M_V g C_R}{N E_M} + \frac{M_V g \sin \theta_{hill}}{N E_M} \qquad (6.10)$$

the speed for the terrain profile with $\theta_{hill} = 0.03789$ *radians* is

$$rpm_{mot}^2 + \frac{0.1055}{0.000000697} rpm_{mot} + \frac{(28.62 - 263.9)}{0.000000697} = 0 \qquad (6.11)$$

which is

$$rpm_{mot} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad (6.12)$$
$$= -153560 \ \ or \ \ 2198.2$$

The motor rpm in (6.12) equates to a vehicle speed of $v_{car} = 26.9$ *mph* or *12.0 m/s*. This value is within *94%* of the simulated value presented in Figure 6-7.

Using a speed of *12.0 m/s*, the theoretical total steady state force acting on the vehicle is

$$F_{vehicle} = F_w + F_R + F_{HG} = M_V g C_R + \frac{1}{2}\rho_A C_D A v^2 + M_V g \sin \theta_{hill}$$
$$= (295.8)(9.8)[(0.0105) + \sin(0.03789)] + \frac{1}{2}(1.2)(0.16)(1.2)(12.0^2)$$
$$= 156.8\,(Newtons)$$

$$(6.13)$$

50
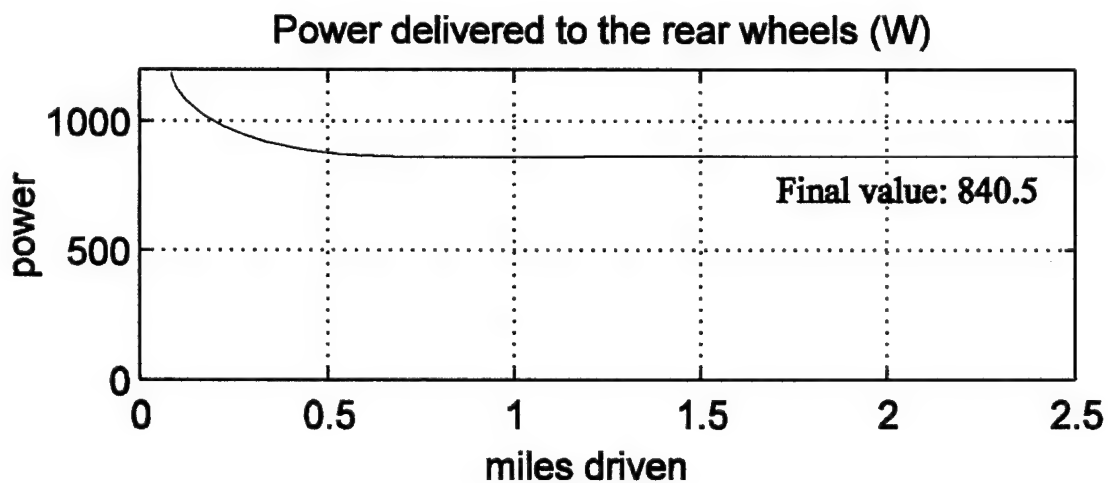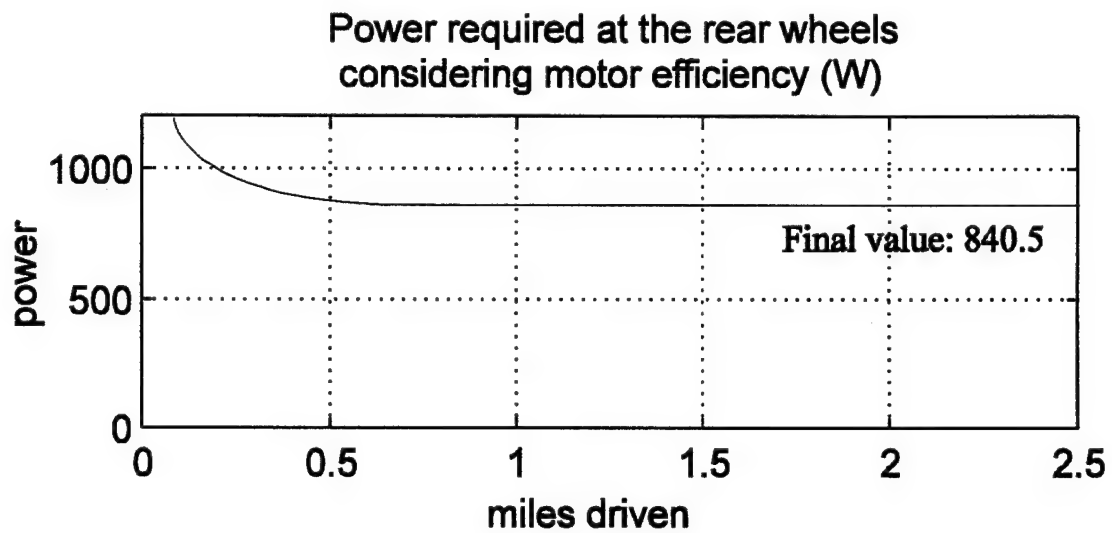
Figure 6-5. Plots of the power required at the rear wheel and power delivered by
the motor for flat terrain driving.
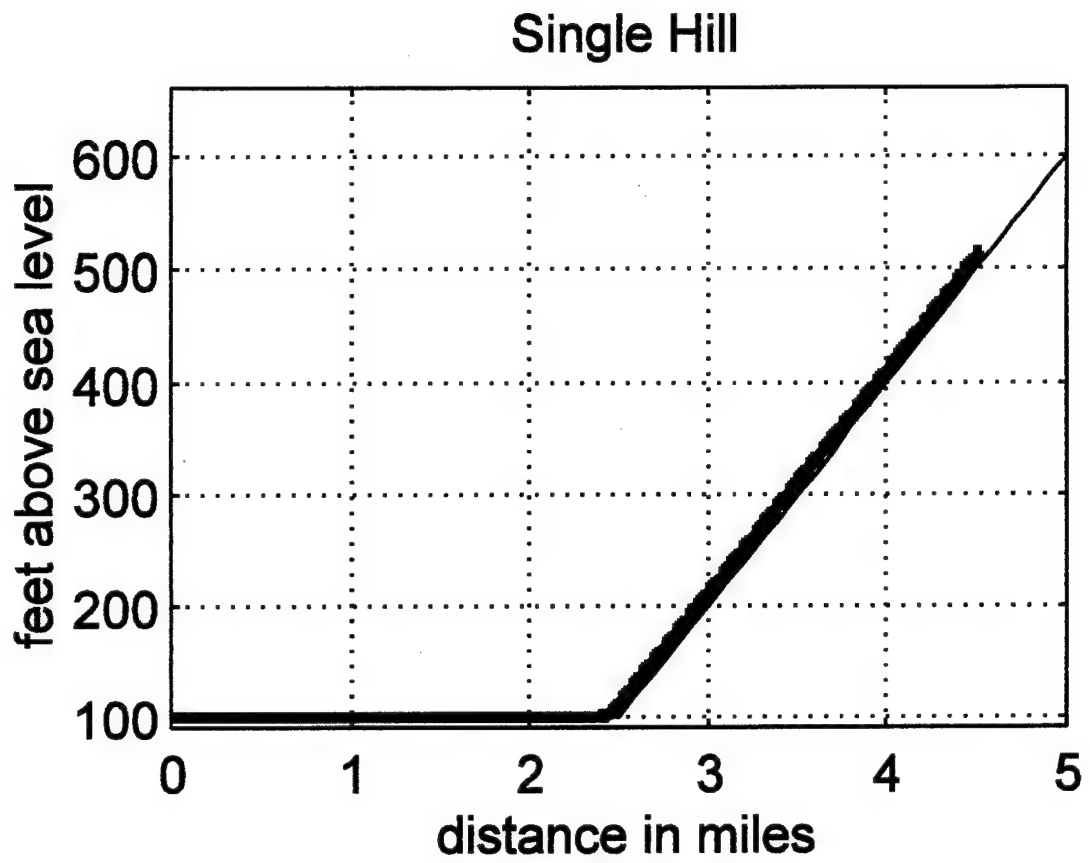
Figure 6-6. Plot of single hill terrain profile .
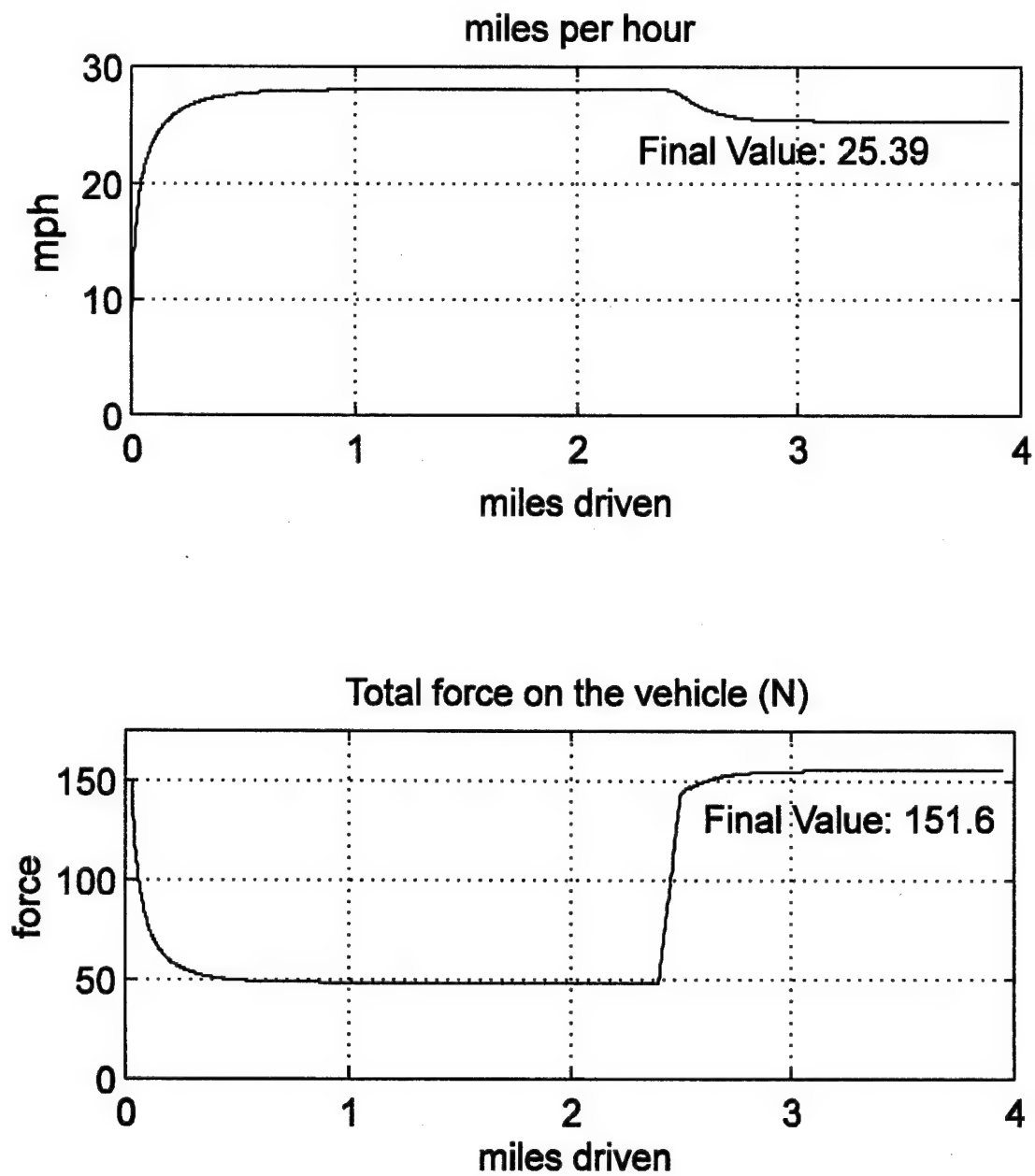
Figure 6-7. Plots of the vehicle speed and the total force acting on the vehicle for a single hill terrain profile.

The simulated force in Figure 6-7 is within *97%* of the computed force in (6.13).

Using equation (4.8), the force from (6.13), and data from Table 6-1, the theoretical torque delivered to the motor for this terrain profile is

$$
\begin{aligned}
T_{load} &= \frac{r(F_w + F_R + F_{HG})}{NE_M} \\
&= \frac{(0.26)(156.8)}{(5)(0.98)} \\
&= 8.32(Newtons - meters)
\end{aligned}
\tag{6.14}
$$

Utilizing (6.12), (6.14), and the efficiency plot in Figure 6-1, the motor efficiency is approximately 85 %. By dividing (6.14) by the motor efficiency, the theoretical load torque delivered to the motor is

$$
T_{motload} = \frac{8.32}{0.85} = 9.78(Newton - meters)
\tag{6.15}
$$

Then utilizing (6.15) and equation (3.24) the theoretical current drawn by the motor is

$$
Idc = \left(\frac{3}{\pi}\right)i_{qs}^r = \left(\frac{3}{\pi}\right)\left(\frac{T_{motload}}{k_t}\right) = \left(\frac{3}{\pi}\right)\left(\frac{9.78}{0.191}\right) = 51.2(Amps)
\tag{6.16}
$$

As seen in Figure 6-8, the theoretical load torque and current are within *92%* of the values obtained from the simulation.

For the power comparison, the theoretical power required for this driving profile is computed from the performance equation (4.2). Utilizing the total force on the vehicle given in (6.13) and the vehicle velocity, the power required at the rear wheels is

$$
P_R = v(F_w + F_R + F_{HG}) = (12.0)(156.8) = 1881.6(Watts)
\tag{6.17}
$$

Dividing (6.17) by the motor efficiency and drive train efficiency, the power required from the motor is

$$
P_{motor} = \frac{P_R}{E_E E_M} = \frac{1881.6}{(0.86)(0.98)} = 2258.5(Watts)
\tag{6.18}
$$

54
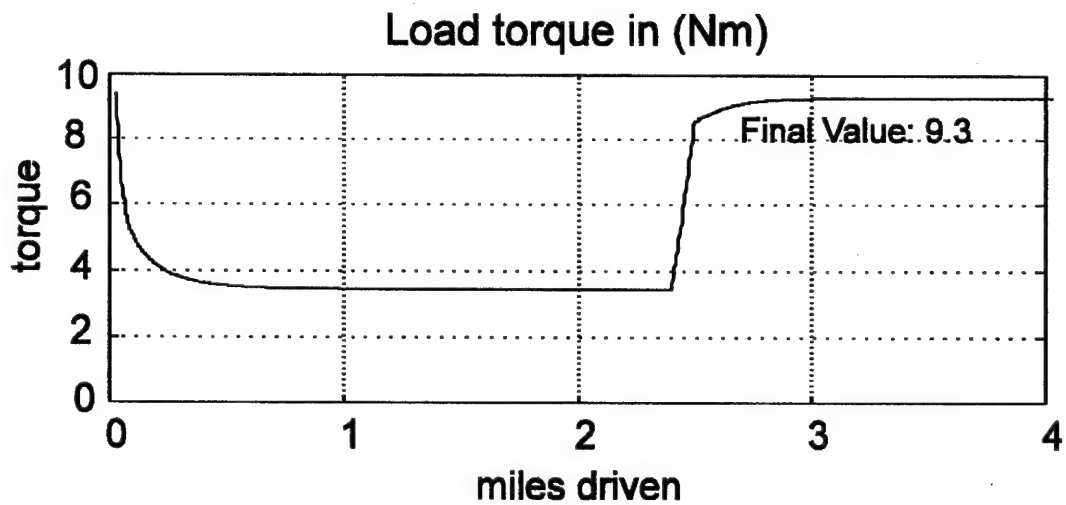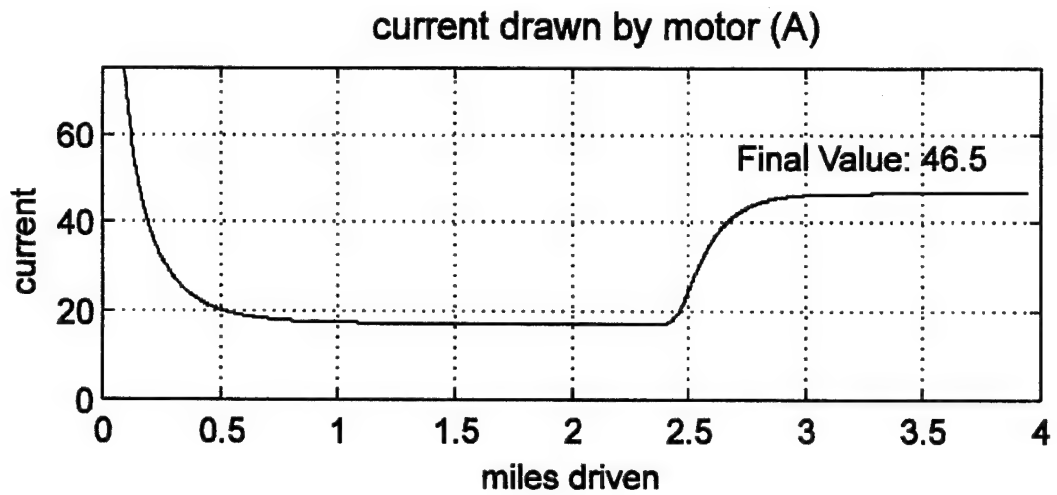
Figure 6-8. Plots of the current drawn by the motor and the load torque delivered to the motor for a single hill driving profile.

The output plots of the power delivered by the motor and the power required at the rear wheels with consideration for system efficiencies are presented in Figure 6-9. The simulated values are within 92% of the expected value of power computed in (6.18). If the vehicle speed had been computed with consideration of motor efficiency, there would have been less variation between theoretical and simulated results.

3. Simulation Driving Over Variable Elevation Terrain

The next simulation is a variable or "hilly" terrain. Solar array current was a constant 11 amps. The terrain profile for this simulation is shown in Figure 6-10.

Since the simulation has been shown to correspond closely with expected values for the flat and single hill driving profiles, the plots for this variable terrain will be presented without theoretical comparison. As seen in Figures 6-11 through 6-13, the outputs are consistent with the results comparisons made in the two previous driving profiles. Observation of Figure 6-13 shows that the power delivered equals the power required. This would indicate that the simulation model does accurately portray an electric vehicle driving over a variable terrain.

Also observed in Figure 6-13 is the regenerative action of the motor when the forces acting against the vehicle motion are overcome by forces which act to propel the vehicle forward. This creates a current regeneration action from the motor which is used by the system to charge the battery, increasing the battery state-of-charge.

4. The Optimum Balance Between Motor Current and Speed

Since the objective of this simulation model is to optimize the design of the vehicle, the next test should be to find the most efficient relationship between current drawn by the motor and speed of the vehicle. This test was driven over flat terrain and the voltage to the motor was stepped in increments of 10 volts using the throttle control. At each voltage, the system was allowed to settle to steady state, then the current being drawn by the motor and the vehicle speed were recorded. Table 6-3 shows the data accumulated from this experiment.

Since the current drawn by the motor is a function of the motor efficiency, there should be a range of speed in which the motor is most efficiently using the energy supplied. By plotting the current as a function of throttle voltage and speed, the most efficient operating points may be easily identified. By examining the output relationship shown in Figure 6-14, the optimum driving speed for this particular vehicle configuration
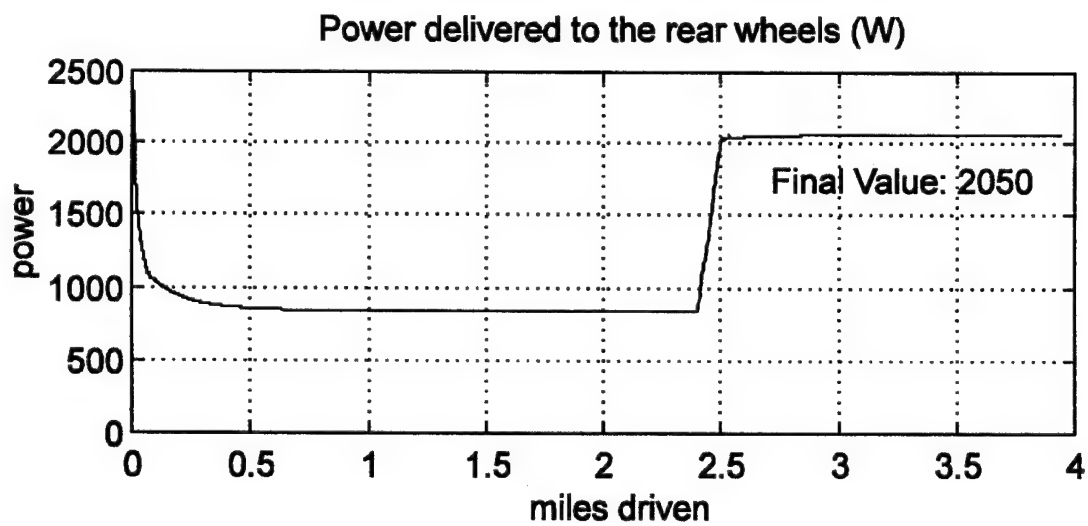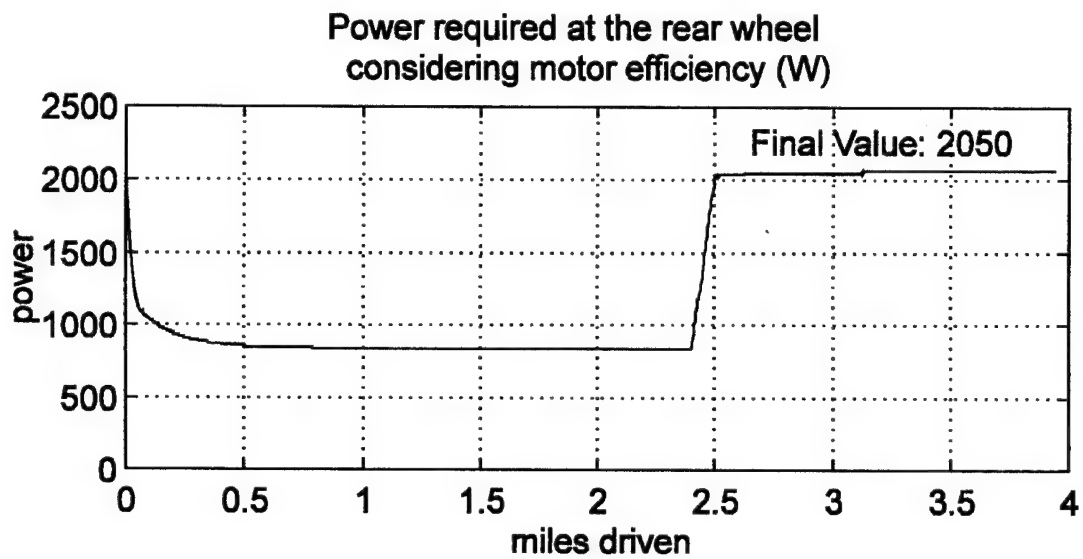
Figure 6-9. Plots of the power required at the rear wheel and the power delivered by the motor for single hill driving profile
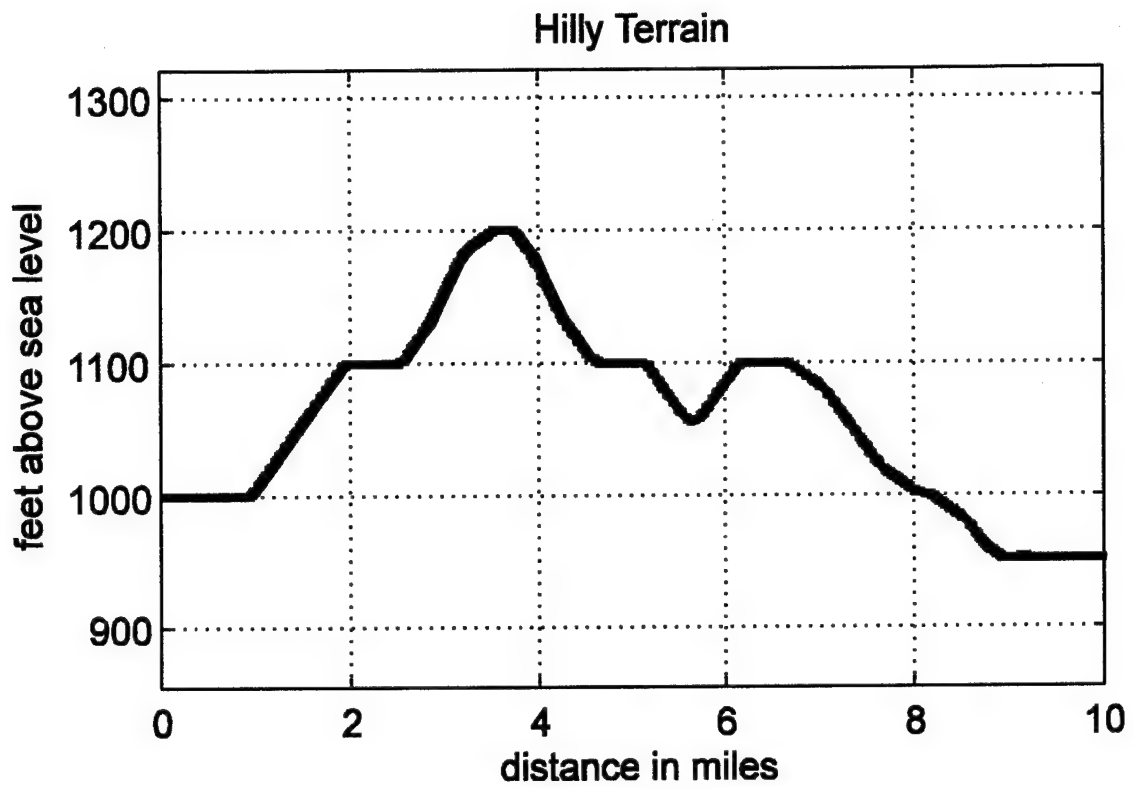
Figure 6-10. Plot of the variable elevation terrain profile.
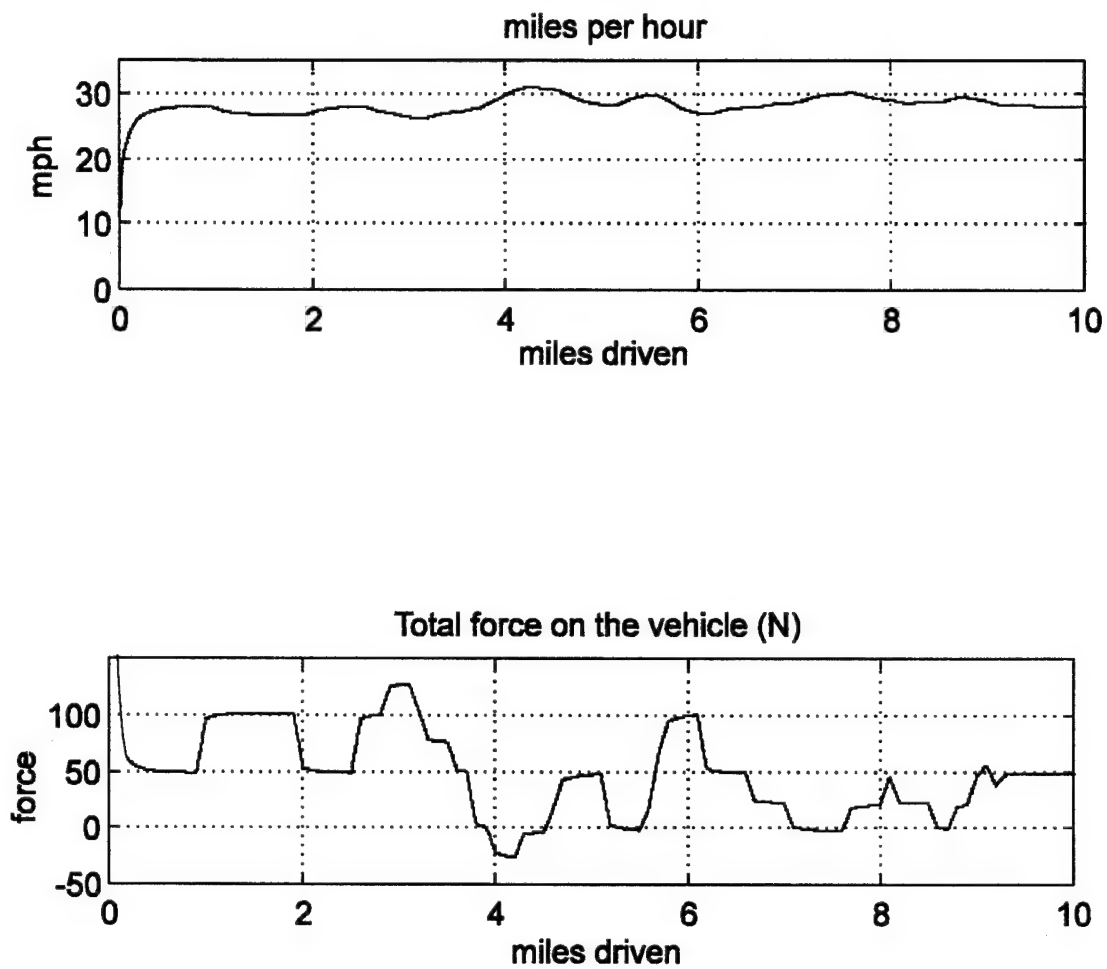
Figure 6-11. Plots of the vehicle speed and the total force acting on the vehicle for a variable terrain profile.
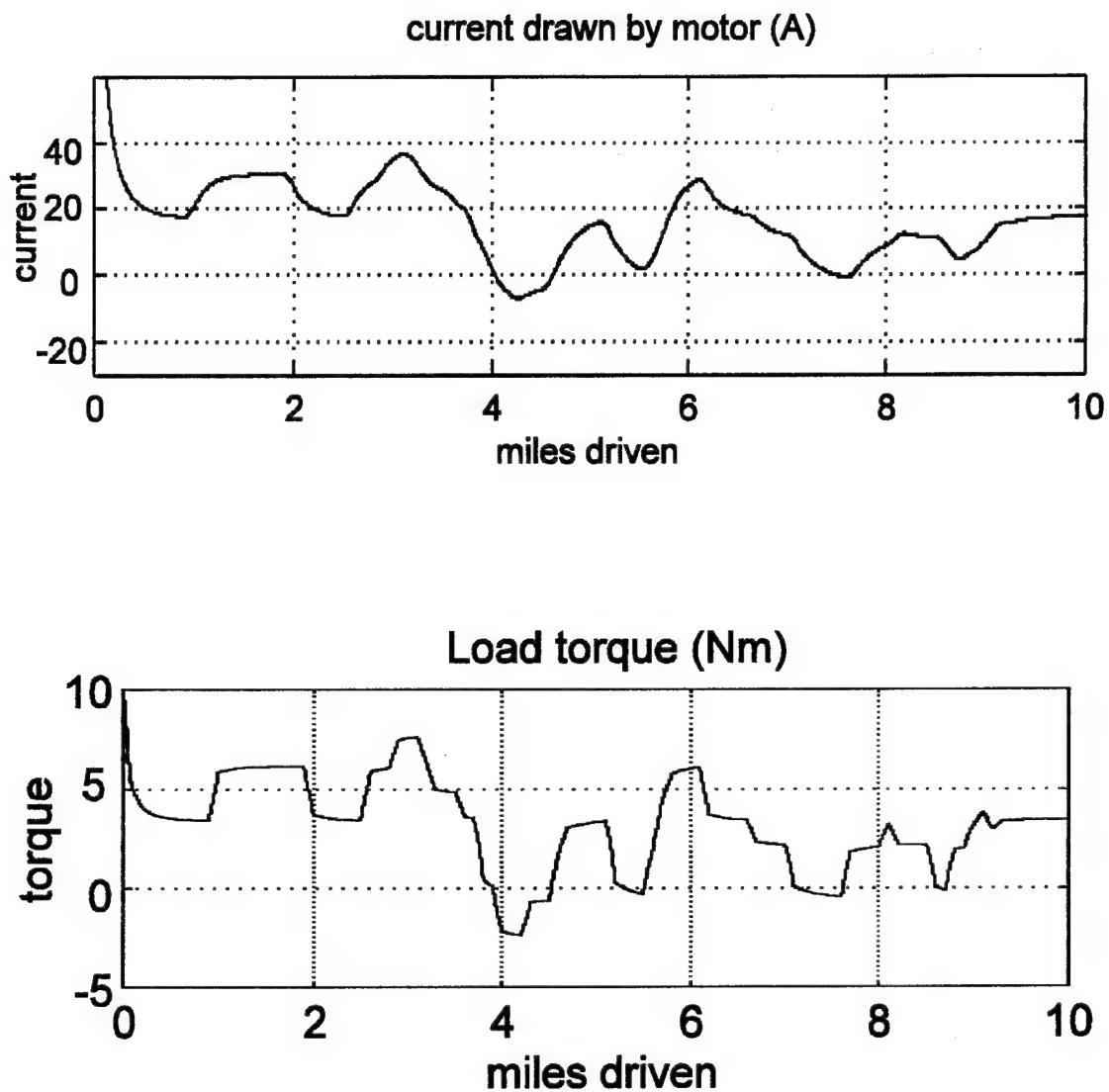
Figure 6-12. Plots of the current drawn by the motor and the load torque delivered to the motor for a variable terrain profile.

60
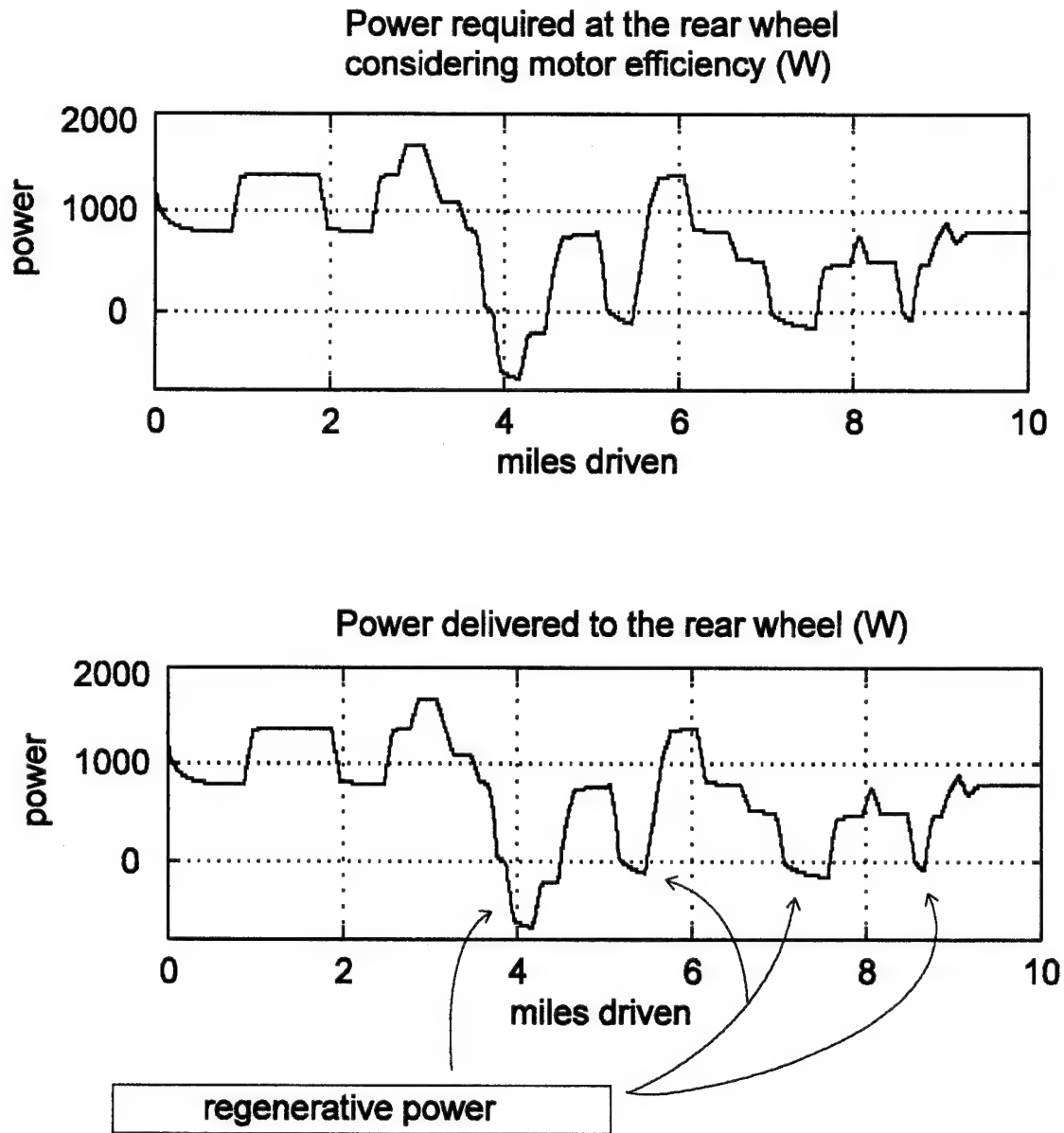
Figure 6-13. Plots of the power required at the rear wheel and the power delivered by the motor for a variable terrain profile.

would be between 25 and 30 mph. This test could also be used to determine the optimum driving speeds for variable terrain's and different gear ratios.

Throttled Voltage (V)	Motor Current(A)	Vehicle Speed (mph)
14	26.5	7.5
24	21.4	13.5
34.2	18.9	19
44.2	17.4	24.9
54.3	17.35	30.2
64.4	18.5	35.5
74.5	20.3	40.1
84.5	22.6	44.9
94.7	24.9	49.1
100	26.1	51.2

Table 6-3. Motor Current and Vehicle Speed for Incremental Voltages.

Efficient vehicle configurations can be achieved by changing the individual vehicle parameters necessary to optimize the current - speed relationship.
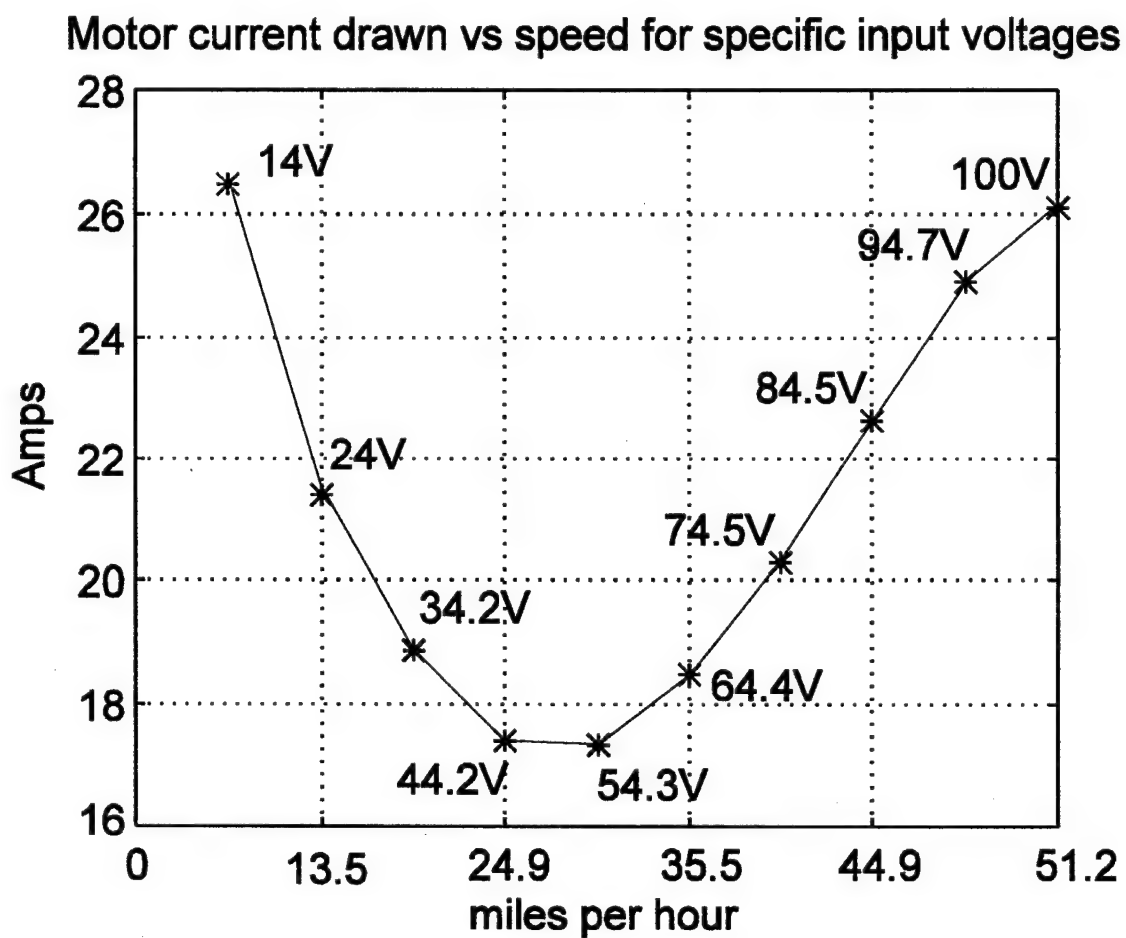
Figure 6-14. Plot of motor current and speed relationship for specific input voltages to the motor.

VII. CONCLUSIONS

There is a need for efficient analysis of alternative energy propulsion systems for the automobiles of tomorrow. By developing systems that quickly analyze a particular vehicle configuration, the system designer can save the time and money required to actually build full scale models. This thesis has provided a method for the efficient evaluation of electricity as an alternative energy source in automobile propulsion.

A. CURRENT APPLICATION

The original goal of this thesis was to model a solar powered electric vehicle for use in design analysis of vehicles configured under the rules of the World Solar Challenge. That task has been successfully completed.

Presentation of this thesis was made to the *Solar Phantom* Team at Rose-Hulman Institute of Technology in Terra Huete, Indiana on March 6, 1995. The system evaluation and output presented in the previous chapters were verified by observations made by the Rose-Hulman team during previous solar powered races in the *Solar Phantom I* and *II*. The team is presently constructing the *Solar Phantom III*, which has a radically different design than the previous two cars. The simulation model developed in this thesis will be used to develop a daily driving strategy for the Solar Phantom Team when they participate in Sunrayce 95 on June 20, 1995. This strategy should assist the team in efficiently using the energy available from the solar array, battery and motor at variable speeds and over variable terrain.

B. FUTURE WORK

This system is by no means complete. Future work could include improvement in the areas of:
- modeling temperature effects on the motor and solar cells
- modeling the motor to account for harmonic losses
- modeling a friction braking mechanism
- adding regenerative current control
- incorporating the effects of auxiliary components in power consumption
- incorporate a random number generator for the cloud cover coefficient to simulate variability of sunlight to the solar array

Other work which could make this system more applicable to industry would be to develop the models for major components such as the battery, motor, vehicle

64

characteristics, and solar array as individual "plants" and evaluate them with a digital Signal Processing and Control Engineering *(dSPACE)* system. The plant models could be built in actual hardware, then connected to the simulation program through the d-Space interface. Through this connection, the simulation program may be used to operate the hardware plant. The systems designer may observe how hardware being considered for use in the vehicle design, reacts under actual operating conditions. The d-Space model would be much more cost efficient for actual component evaluation than construction of full scale models.

APPENDIX A. SIMULATOR USER INSTRUCTIONS

1. The Solar/Electric Vehicle Simulator will only run from a Matlab control window. The specific version of Matlab is at least 4.1 with Simulink version 1.3c. Ensure that these are loaded on your system before attempting to run the simulator.

2. Once in the Matlab control window, type - **solar**. The user control window and the default terrain profile figures should appear in the upper right corner of the monitor. Figure A-1 shows the user control window for the simulator. ·
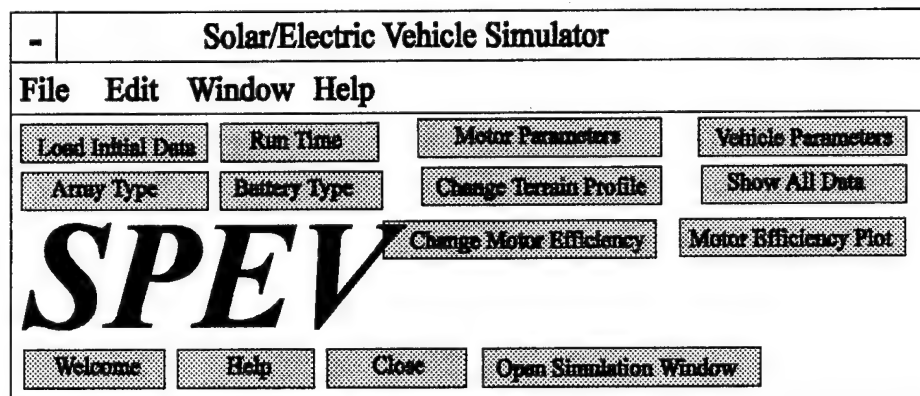
Figure A-1. SPEV simulation window

Figure A-2 shows the default driving terrain profile loaded upon simulator system initialization. Each button in the user control window invokes a specific function. Theses functions are explained below

   a) Load Initial Data: Loads the initial default values for all electrical and mechanical parameters. Once loaded, all values are displayed in the Matlab control window.

   b) Run Time: Modifies the total simulation time, time of day to start driving, and integration time step.

   c) Motor Parameters: Allows the user to modify or change the motor characteristics, or implement a new motor. Input the requested data at the prompts in the Matlab control window.

66

d) Vehicle Parameters: Allows any of the vehicles mechanical characteristics to be modified. Input requested data at the prompts in the Matlab control window.
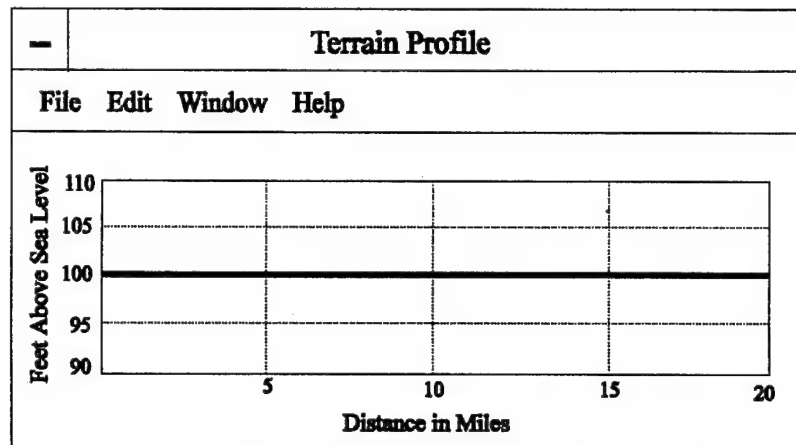
Figure A-2. Default driving terrain profile

e) Array Type: Change between tracking, flat, or curved solar array configurations and specify the peak array current.

f) Battery Type: Change between Lead-Acid or Silver-Zinc battery packs.

g) Terrain Profile: Allows the user to choose one of three terrain profiles. At the prompt in the Matlab control window choose flat, single hill, or load a file of variable terrain data. If the variable terrain is chosen, the data must be entered from a "filename.dat" text file with two columns of information. Column one has distance in increments of .1 miles and column two contains the corresponding elevation in feet above sea level. Each time a terrain profile is changed, a modified terrain profile window is displayed.

h) Show All Data: Provides an updated display of all important parameters loaded into the simulation program. This function useful in verifying changes in system parameters.

i) Motor Efficiency Plot: Provides a 3-dimensional plot of the motor efficiency as provided by the manufacturer. The user must supply the efficiency matrix for the specified motor. The columns correspond to increasing load torque, while the rows correspond to increasing motor rpm.

j) Welcome: Shows a brief welcome to the simulator.

67

k) Help: Provides a brief help on running a simulation from the simulation window.

l) Close: Closes the user window.

m) Open Simulation Window: Opens the primary vehicle simulation window. Once all previous parameter input requirements have been meet, simply click on the *"Simulation - Start"* button.

3. Once the simulation has begun, the data output window shown in figure A-3 will be displayed at the bottom left of the monitor.

▬	Output
File Edit Window Help	

Mph

Motor Current (amps):   Array Current (amps):   RPM: 15
●   Run time (secs):   Hill Grade (%):

Ahrs Used: Odometer (miles):
SOC (%):   Speedometer (mph):   0   30
Battery Voltage (volts):
Vehicle Load (N-m):   System Efficiency (%):

45
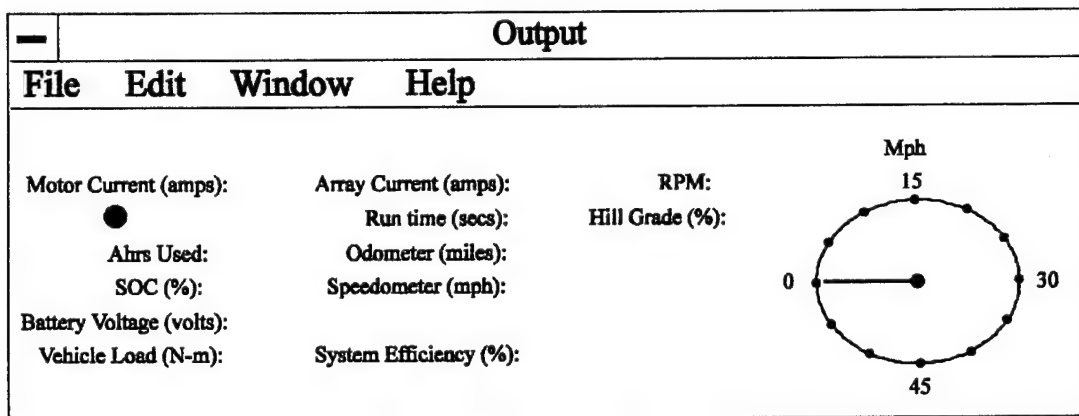
Figure A-3. Simulation On-Line Display window

The output indicators are self-explanatory and will provide continuously updated data display throughout the simulation.

4. The simulation may be stopped at any time simply by clicking on the *"Simulation - Stop"* button in the main Simulink simulation window.

5. By double-clicking on the *"Plots"* box in the Simulink simulation window, plots of the important vehicle characteristics, as well as average values of output data will be displayed in four separate figure windows. Each of these windows may be closed by clicking on the *"File - Close"* indicator.

A. FUNCTION SOLAR.M FOR MAIN PROGRAM

```
%      Solar Powered Electric Vehicle Simulator
% Masters Thesis EE Naval-Post Graduate School, Monterey
% This system will simulate a solar/electric powered vehicle
% for any type of vehicle mechanical or electric characteristics.
%
%      This system may be used to optimize vehicle configurations.
% each function is explained separately...
%
%      This function initiates the system user control window and
% the default terrain profile.
%
%      This system was developed by Steven Roerig, LT, USN
% Date: 5 Mar 1995....all rights reserved...
%
%*******************************************
%      Create the Efficiency matrix (call efffunc)
% initialize the all data matrices and set default terrain
% to flat.

res=10;              % 10 times size of efficiency matrix
nmot = 0;                % initialize matrix
[effmat rpmin Tlin nmot isnew]=efffunc(res,nmot);

datamat = zeros(3,23);                          % parameter initialization
strmat = ['                    ';'                    '];
hildef=0;
[distvec hillang initelev hillvec hildef]=hillfunc(hildef);


%********************************
% Initialize the user control window and set the position
%
fignumber=figure( ...
'Position',[0 0 570 200], ...
'Color',[0 0 1], ...
'NumberTitle','off', ...
'Name','Welcome to the Solar/Electric Vehicle Simulator', ...
'Resize','off', ...
'Pointer','arrow');

axes('Units','normalized', ...
```

```
'Position',[.15 .25 .75 .45]);
plot([0 2 2.2 .2 .4 2.4],[0 0 .1 .1 .2 .2],'m', ...
    [3 3.4 5.4 5.2 3.2],[-.01 .2 .2 .1 .1],'m', ...
    [8 6 6.2 7.4],[0 0 .1 .1],'m',[6.2 6.4 8.4],[.1 .2 .2],'m', ...
    [9.4 9 11],[.21 -.01 .22],'m', ...
    'LineWidth',12)
axis([-.5 11.5 -.1 .3])
axis off


%*********************************
% Create the OPEN SIMULATION WINDOW Button
%
spevbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[320 10 160 30], ...
'String','Open Simulation Window', ...
'Visible','off', ...
'Enable','off', ...
'Callback','newsim');


%*********************************
% Create WELCOME Button
%
titl1 = 'Welcome';
welcscrn =[' * Welcome to the Solar Vehicle Simulator.   '
        ' * Please make your selections from the      '
        ' * Control Buttons in the window and follow  '
        ' * the directions at each prompt in the      '
        ' * Matlab Command window.                    '
        ' *                                           '
        ' *                                           '
        ' *                                           '
        ' *          -- Enjoy the ride! --            '];


welcbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[5 10 100 30], ...
'Visible','off', ...
'String','Welcome', ...
'Enable','off', ...
'Callback','helpsp(titl1,welcscrn)');


%*********************************
```

```
% Create the HELP Button
%
titl2 = 'Solar/Electric Vehicle Simulator Help';
helpscrn = ['                                    '
            ' The SPEV Simulator will assist you in    '
            ' optimzing the configuration of your      '
            ' vehicle by providing user controlled     '
            ' parameter variation and continuous       '
            ' display.  Your vehicle characteristics   '
            ' may be easily changed without the initial'
            ' design costs.                      '
            '                                    '
            ' Use the "User Control Buttons" to input  '
            ' the vehicle characteristics, then change '
            ' individual characteristics by following  '
            ' the directions in the Matlab Control      '
            ' Window.                            '
            '                                    '
            ' Click on "Open Simulation Window" to open'
            ' the Simulink model for the vehicle.      '
            '                                    '
            ' Use the "Throttle" to adjust the vehicle '
            ' speed and "Cloud Cover" to adjust the     '
            ' current from the solar array (simulate    '
            ' cloud cover) ... see the results.        '];


helpbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[110 10 100 30], ...
'Visible','off', ...
'String','Help', ...
'Enable','off', ...
'Callback','helpsp(titl2,helpscrn)');


%********************************
% Create the CLOSE Button ... closes the user control window

closebut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[215 10 100 30], ...
'Visible','off', ...
'String','Close', ...
'Enable','off', ...
```

```
'Callback','close(gcf)');

%********************************
% Create all User Control Buttons


%    LOAD INITIAL DATA ... loads all initial and default data for
% first run simulations.
% The default is mechanical data from UC Berkeley's solar car
% with a Unique Mobility 11.3 Hp motor, tracking array, and
% lead-acid batteries.
%
firstbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[5 170 140 30], ...
'Visible','off', ...
'String','Load Initial Data', ...
'Enable','off', ...
'Callback','[datamat strmat]=spevfunc("first",datamat,strmat);');


%    RUN TIME ...  controls the total run time desired, the time
% of day to start simulation, and the integration time step (dt)
% which controls the speed of simulation.
%
runbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[150 170 100 30], ...
'Visible','off', ...
'String','Run Time', ...
'Enable','off', ...
'Callback','[datamat strmat]=spevfunc("runtime",datamat,strmat);');


%    MOTOR PARAMETERS... allows the user to change the motor type or
% individual parameters ... default is a 11.3Hp/8.4KW motor.
%
motbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[255 170 145 30], ...
'Visible','off', ...
'String','Motor Parameters', ...
```

```
'Enable','off', ...
'Callback','[datamat strmat]=spevfunc("motor",datamat,strmat);');


%    VEHICLE PARAMETERS ... allows the user to change the vehicles
% mechanical parameters.  The default data is obtained from the
% UC Berkeley (CALSOL) team vehcile.
%
vehbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[405 170 155 30], ...
'Visible','off', ...
'String','Vehicle parameters', ...
'Enable','off', ...
'Callback','[datamat strmat]=spevfunc("vehicle",datamat,strmat);');


%    ARRAY TYPE ... allows the user to choose between tracking,
% flat, or curved array configurations and specify the peak current
% expected. Default is tracking @ 11 amps peak
%
araybut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[5 135 140 30], ...
'Visible','off', ...
'String','Array Type', ...
'Enable','off', ...
'Callback', '[datamat strmat]=spevfunc("saray",datamat,strmat);');


%    BATTERY TYPE ... choose between lead-acid or silver-zinc batteries.
% battery voltage is set to maintain a 100 volt bus voltage.
% Default is lead-acid battery.
%
battbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[150 135 100 30], ...
'Visible','off', ...
'String','Batt Type', ...
'Enable','off', ...
'Callback','[datamat strmat]=spevfunc("battery",datamat,strmat);');
```

```
%   CHANGE TERRAIN PROFILE... allows user the change the driving terrain.
% Choose between flat, single hill, or terrain profile from a data file.  Data
% must be loaded from a 2 column "filename.dat" text file.  First column
% is distance from start in 1/10 mile increments and the corresponding
% elevation is feet above sea level. Default is flat terrain.
%
hillbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[255 135 145 30], ...
'Visible','off', ...
'String',' Change Terrain Profile', ...
'Enable','off', ...
'Callback','[distvec hillang initelev hillvec hildef]=hillfunc(hildef);');


%   SHOW ALL DATA ... shows all significant data into simulator.
% Useful after making changes.
%
showbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[405 135 155 30], ...
'Visible','off', ...
'String','Show All Data', ...
'Enable','off', ...
'Callback','[datamat strmat]=spevfunc("showall",datamat,strmat);');


%   MOTOR EFFICIENCY PLOT ... 3-dimensional plot of motor efficiency.
% Each motor must have an efficiency matrix.
%
effbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[405 100 155 30], ...
'Visible','off', ...
'String','Motor Efficiency Plot', ...
'Enable','off', ...
'Callback','effpltf(rpmin,Tlin,effmat,isnew);');


%   CHANGE MOTOR EFFICIENCY ... change motor efficiency.
% Each motor must have an efficiency matrix.
```

```
%
effmotbut = uicontrol( ...
'Style','pushbutton', ...
'Units','pixels', ...
'Position',[235 100 165 30], ...
'Visible','off', ...
'String','Change Motor Efficiency', ...
'Enable','off', ...
'Callback','[effmat rpmin Tlin nmot isnew]=efffunc(res,nmot);');

% Turns on all buttons ....
%
set([welcbut spevbut helpbut closebut firstbut runbut motbut ...
    vehbut araybut battbut hillbut showbut effbut effmotbut], ...
    'Enable','on', ...
    'Visible','on');
```

B. FUNCTION SPEVFUNC.M TO CHANGE SYSTEM PARAMATERS

```
%   Solar Powered Electric Vehicle Simulator.
% Developed by Steven J. Roerig, LT, USN, at the Naval-Post Graduate
% School, Monterey, Ca.   Date: 7/26/94
%
%   This function will change the system paramters when called.
%

function [spevdata,strdata] = spevfunc(action,data,strmat);

% Load the initial data and vehicle parameters...
%
if strcmp(action,'first');
  disp('Since this is your initial run, the default data has been entered')
  disp('for a 11.3 Hp, tracking array, Pb-Acid vehicle');
  data(1,(1:10)) = [.5 3600*9 9 24 0 .014 .036 25e-6/.036 20 .184];
  data(1,(11:23)) = [.16 .0105 1.2 295.8 .98 35 100 0 0 .26 5 0 0];
  data(2,:) = 1.1*[10 10 10 10 10 10 10 10 10 10 ...
        10 10 10 10 10 10 10 9.9 9.8 0 0 0 0];
  strmat(1,:) = 'Tracking             ';
  data(3,:) = 60*([.6 .9 1.25 1.4 1.55 1.65 1.7 1.75 1.8 1.84 1.88 ...
        1.91 1.94 1.96 1.98 2.0 2.02 2.04 2.06 2.07 2.08 0 0]);
  strmat(2,:) = 'Pb-Acid             ';
end


% Modify the run time parameters.
%
```

```matlab
if strcmp(action,'runtime')
   disp('Note: All run times will begin at 9 AM.  Maximum run time')
   disp('is 8 Hours.')
   tmin = input('     Enter length of simulation run (minutes): ');
   data(1,2) = 60*tmin;                              %final time in seconds
   timein = input('     Simulation start time (btwn 0900 and 1800): ');
   data(1,3) = timein/100;
   data(1,1) = input('     Time step (how slow/fast to run: btwn .0001 and 1 sec): ');
end


%   Modify the motor characteristics
%
if strcmp(action,'motor')
disp('Motor Characteristics -')
   data(1,4) = input('  Number of poles (P): ');
   data(1,5) = input('  Damping coefficient (B): ');
   data(1,6) = input('  Inertia (J): ');
   data(1,7) = input('  Winding resistance (ra): ');
   data(1,8) = input('  Electrical time-constant (ta): ');
   data(1,9) = input('  Back-EMF constant (ke): ');
   data(1,10) = input('  Torque constant (kt): ');
end


%   Modify the vehicle characteristics
%
if strcmp(action,'vehicle')
disp('Vehicle Characteristics -')
   data(1,11) = input('  Drag coefficient (Cd): ');
   data(1,12) = input('  Rolling resistance coefficient (Cr): ');
   data(1,13) = input('  Frontal area (A in m^2): ');
   data(1,14) = input('  Vehicle mass (Mv in kg): ');
   data(1,20) = input('  Drive wheel radius (r in meters): ');
   data(1,21) = input('  Drive ratio (belt, chain, or gear - 1:N) (N=): ');
   effm = input('  Drive train efficiency (%): ');
   data(1,15) = effm/100;
end


%   Change solar array type and peak output current.
%
arr_dat = [10 10 10 10 10 10 10 10 10 10 ...
          10 10 10 10 10 10 10 9.9 9.8 0 0 0 0];
if strcmp(action,'saray')
```

76

```matlab
      ar_choice = input('Solar Array type: Flat(1),Curved(2),Tracking(3): Number - ');
      peak = input('What is your expected peak current: ');
        if ar_choice == 1
          data(2,:) = (peak/10)*[.72 .78 .83 .88 .92 .97 .98 .98 .99 .985 .98 ...
                  .95 .92 .90 .85 .82 .77 .67 .48 0 0 0 0].*arr_dat;
          strmat(1,:) = 'Flat             ';
        elseif ar_choice == 2
          data(2,:) = (peak/10)*[.59 .63 .68 .75 .79 .85 .87 .87 .88 .87 ...
                  .86 .85 .82 .78 .72 .69 .64 .55 .48 0 0 0 0].*arr_dat;
          strmat(1,:) = 'Curved           ';
        elseif ar_choice == 3
          data(2,:) = (peak/10)*arr_dat;
          strmat(1,:) = 'Tracking         ';
        end
end


%   Change battery type.
%
if strcmp(action,'battery')
   bat_choice = input('Battery type: Pb-Acid(1),Ag-Zn(2): Number - ');
     if bat_choice == 1
        data(3,:) = 60*([.6 .9 1.25 1.4 1.55 1.65 1.7 1.75 1.8 1.84 1.88 ...
        1.91 1.94 1.96 1.98 2.0 2.02 2.04 2.06 2.07 2.08 0 0]);
        strmat(2,:) = 'Pb-Acid          ';
     elseif bat_choice == 2
        data(3,:) = 68*([.8 1.35 1.45 1.47 1.48 1.49 1.495 1.497 ...
        1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.5 1.55 1.7 1.8 1.85 1.9 0 0]);
        strmat(2,:) = 'Ag-Zn            ';
     end
   data(1,16) = input('What is the A-h rating of your battery: ');
   data(1,17) = input('What is your initial state of charge (percent): ');
end


%  Show all data upon initialization and whenever called.
%
if strcmp(action,'showall') | strcmp(action,'first')
  disp('')
  disp('The below information is currently is loaded into the simulator:')
  disp(['  Run time: ',num2str(data(1,2)/60),' minutes'])
  disp(['Start time: ',num2str(data(1,3))])
  disp(['  Time step: ',num2str(data(1,1))])

  disp(' ')
```

```
disp('Motor Characteristics ')
disp(['    Number of poles: ',num2str(data(1,4))]);
disp(['    Damping coefficient: ',num2str(data(1,5))]);
disp(['    Inertia: ',num2str(data(1,6))]);
disp(['    Armature resistance: ',num2str(data(1,7))]);
disp(['    Electrical time-constant: ',num2str(data(1,8))]);
disp(['    Back-EMF constant: ',num2str(data(1,9))]);
disp(['    Torque constant: ',num2str(data(1,10))]);

disp(' ');
disp('Vehicle Charcteristics ');
disp(['    Drag coefficient: ',num2str(data(1,11))]);
disp(['    Rolling resistance coefficient: ',num2str(data(1,12))]);
disp(['    Frontal area (meters sqrd): ',num2str(data(1,13))]);
disp(['    Vehicle mass (kilo-gram): ',num2str(data(1,14))]);
disp(['    Drive wheel radius (meters): ',num2str(data(1,20))])
disp(['    Drive ratio (N): 1:',num2str(data(1,21))]);
disp(['    Drive train efficiency: ',num2str(data(1,15)*100),'%']);

disp(' ');
disp(['Solar Array Type: ',strmat(1,1:21)]);

disp(' ');
disp(['Battery Type: ',strmat(2,1:21)]);

disp(' ')

disp('If you need to make changes in your data, just click on the')
disp('item in the Menu Bar.')
end


% Output to the system.
%
spevdata = [data(1,1:23);data(2,1:23);data(3,1:23)];
strdata = [strmat(1,1:23);strmat(2,1:23)];
    % end of function
```

C. FUNCTION HILLFUNC.M TO CHANGE TERRAIN PROFILE

```
%   This function performs all hill and terrain profile modifications
% when called from the user control window.
%

function [distvec,hillang,initelev,hillvec,hildef]=hillfunc(hildef);
```

```matlab
if hildef ~= 0
  whichone = input('For a single hill enter - 1, hill profile from file enter - 2, or flat driving
enter - 3: ');
  disp('')
else
  whichone = 3;
end


%   For a single hill terrain profile.
%
if whichone == 1
  st1 = input('How many miles is the hill from simulation start? ');
  st2 = input('How long is the hill (in miles)? ');
  el1 = input('What is your initial evelation (feet)? ');
  el2 = input('What is will be your final elevation? ');
  distvec = 0:.1:st1+st2;
  if (el2-el1) > 0
    hillvec = [el1*ones(size(1:length(0:.1:(st1-.1)))) el1:((el2-el1)/(10*st2)):el2];
  elseif (el2-el1) < 0
    hillvec = [el1*ones(size(1:length(0:.1:(st1-.1)))) el1:-((el2-el1)/(10*st2)):el2];
  end
  type = 'Single Hill';

% or for a flat driving profile
%
elseif whichone == 3
  if hildef ~= 0
   st1 = input('How many miles will you be driving? ');
   el1 = input('At what elevation will you be driving (feet)? ');
  else
   st1 = 20;
   el1 = 100;
  end
   distvec = 0:.1:st1;
   if el1 ~= 0
     hillvec = [el1*ones(size(1:length(0:.1:st1)))];
   else
     hillvec = [zeros(size(1:length(0:.1:st1)))];
   end
   type = 'Flat Terrain';
   hildef = 1;

% or if a terrain profile is to be loaded from a text file.
%
```

```
elseif whichone == 2
      disp('NOTE: The input file must be set up with two column vectors. The first');
      disp('column contains the distance from start of simulation in tenths of miles,');
      disp('and the second column contains the elevation above sea level in feet. The');
      disp('file must be a text file of the form "filename.dat".');
      disp('')
      filename = input('Input the filename: ','s');
      fid=fopen(filename);
      dat = fscanf(fid,'%g    %g',[2 inf]);
      fclose(fid);
      distvec= dat(1,:);
      hillvec= dat(2,:);
      type = 'Hilly Terrain';
end

% compute the hill angle (alpha) and output to system
%
initelev=hillvec(1,1);
  for i=2:length(hillvec)
    hillang(i-1)=asin((hillvec(i)-hillvec(i-1))/528);
  end
hillang(length(hillvec))=hillang(length(hillang));

% Plot the updated terrain profile for use during simulation
%
Figures = get(0,'Children');
  for i = 1:length(Figures)
        if strcmp(get(Figures(i),'Type'), 'figure')
        if strcmp(get(Figures(i), 'Name'), 'Terrain Profile')
          close(Figures(i))
          break;
        end
      end
  end

figure('NumberTitle','off','Name','Terrain Profile', ...
          'NextPlot','add','Position',[0 0 450 300]);
plot(distvec,hillvec),grid,ylabel('feet above sea level')
axis([0 max(distvec) (min(hillvec)-(.1*min(hillvec))) (max(hillvec)+(.1*max(hillvec)))])
xlabel('distance in miles'),title(type)
```

D. FUNCTION EFFFUNC.M TO EXPAND SIZE OF EFFICIENCY MATRIX

```
%  This function will expand the efficiency matrix in order
%  to decrease interpolation errors.
%
```

80

```matlab
function [efficiency,rpmin,Tlin,nmot,isnew] = efffunc(N,nmot);

%   Efficiency for the 11.3 Hp motor
%

isnew = 127;
if nmot == 1
  isnew=input('Is this the 127 or 86 Unique motor? ');
end

if isnew == 86
  effmat=[35  55  67  68  69  70  71  70  63  55;
      58  68  72  74  75  74  72  70  68  66;
      68  74  78  78.5 78.6 76  74  72  70  69;
      72  77  82  82  81.9 81  80  77  73  70;
      74  79  84  84.4 85  84.8 83  82  80  78;
      78  80  86  87.3 87.2 87.1 86.5 84.5 83  81.5;
      81  84  89  89.7 90.1 89.8 88  87  86  85.5;
      82  86  90.5 92  92.2 92.8 91  89.2 88.5 87;
      83  87  91.7 94  93.5 92.5 91.6 89.8 88.6 87.1;
      84  87.5 93  94.5 94.2 93  91.9 90  89.5 88.5];
elseif isnew == 127
  effmat=[10  20  32  39  38  40  51  41  28  18;
      18  36  60  71  70  69  72  63  55  45;
      20  50  76  85.5 82  80.5 79  75  71  64;
      30  60  85.5 86  85.5 84.8 83  82  79  74;
      33  71  86  87.5 86.5 86.5 85.5 85  83.5 81;
      38  80  87  88.5 89  88  87  86  95.5 84;
      40  84  89  90  90.5 90  89  88  87  86;
      50  85  89.5 91  92  91.5 91  90  90.5 89.5;
      50  86  90  92  93  93.5 91  91.5 90.5 89.5;
      48  86.5 90.5 92.5 94  93.5 93  92  91  90.5];
  nmot = 1;
end

[n m]=size(effmat);

%*****************************************************
% Increases the size of the matrix from n x m to N*n x N*m
% with linspace interpolation.
%
for j=1:n
  for i=2:m
    eftemp(i-1,:)=linspace(effmat(j,i-1),effmat(j,i),N);
```

81

```
    end
  [c d] = size(eftemp);
   for a=1:c
     for b=1:d-1
        neweffmat(j,(((N-1)*(a-1))+b)) = eftemp(a,b);
     end
   end
end


new2=neweffmat';
[n m]=size(new2);

for j=1:n
  for i=2:m
     eftemp2(i-1,:)=linspace(new2(j,i-1),new2(j,i),N);
  end
  [c d] = size(eftemp2);
   for a=1:c
     for b=1:d-1
        neweffmat2(j,(((N-1)*(a-1))+b)) = eftemp2(a,b);
     end
   end
end


% ********************************************************

efficiency = neweffmat2';
[n m]=size(efficiency);

%  Corresponding torque and rpm vectors.
%
if isnew == 127    %  Unique DR127/CR10
  rpmin=linspace(25,5000,n);
  Tlin=linspace(.1,20.25,m);
elseif isnew == 86  %  Unique DR086/CR10
  rpmin=linspace(200,5600,n);
  Tlin=linspace(.4,15.2,m);
end
```

E. FUNCTION EFFPLTF.M FOR SURFACE PLOT OF EFFICIENCY MATRIX

```
%   This function performs a surface plot of the motor
% efficiency

function effpltf(rpmin,Tlin,effmat,isnew);
```

```
figure('NumberTitle','off','Name','Efficiency Window')
colormap(jet)
surf(Tlin,rpmin,effmat),grid
axis([0 max(Tlin) 0 max(rpmin) 0 100])
shading flat
colorbar
xlabel('Torque'),
ylabel('RPM'),
zlabel('Eff(%)'),
if isnew == 127
    title('Motor Efficiency for Unique DR127/CR10');
elseif isnew == 86
    title('Motor Efficiency for Unique DR086/CR10');
end
```

F. FUNCTION SPEVAN2.M FOR ON-LINE DISPLAY WINDOW

```
function [sys,x0]=spevan2(t,x,u,flag,figname1);

%  spevan2 S-function for animating the on-line display of dynamic
%  output.  To see the specific input corresponding to the parameter
%  u[], open the "On-Line Display" block in the simulation window.
%  Steven Roerig. 7/25/9

global HNDL1
if abs(flag) == 2
    if (x(1) ~= Inf)    % screen has been initialized
         hndls=get(x(13),'UserData');

    if u(5) > -.01 & u(5) < .01      % set output to zero
       set(hndls(1),'String',num2str(0.00));
    else
       set(hndls(1),'String',num2str(u(5)));
    end

         set(hndls(2),'String',num2str(u(3)));
         set(hndls(3),'String',num2str(u(4)));

    if u(1) > -.001 & u(1) < .001    % set output to zero
       set(hndls(4),'String',num2str(0.00));
    else
       set(hndls(4),'String',num2str(u(1)));
    end

       set(hndls(5),'String',num2str(u(2)));
```

```matlab
        set(hndls(8),'String',num2str(u(7)));
        set(hndls(12),'String',num2str(u(6)));
          set(hndls(9),'String',num2str(u(8)));


   if u(9) > -.001 & u(9) < .001    % set vehicle "stopped"
        set(hndls(10),'String',num2str(0.00));
        set(hndls(11),'String','Stopped.','Color',[1 1 0]);
   elseif u(9) < -.001         % set vehcile "backward"
        set(hndls(10),'String',num2str(u(9)));
        set(hndls(11),'String','Going Backward!!','Color',[1 0 0])
   else                        % set vehicle "going forward"
        set(hndls(10),'String',num2str(u(9)));
        set(hndls(11),'String','Going Forward.','Color',[0 1 1])


   end


   if u(5) > .001           % set drawing current
        set(hndls(6),'XData',1,'YData',2.5,'Color',[0 1 0]);
        set(hndls(7),'String','using');
   elseif u(5) < -.001       % set regenerating current
        set(hndls(6),'XData',1,'YData',2.5,'Color',[1 0 0]);
        set(hndls(7),'String','regen');
   else                     % set motor stopped
        set(hndls(6),'XData',1,'YData',2.5,'Color',[0 0 0]);
        set(hndls(7),'String',' ');
   end


        set(hndls(13),'String',num2str(u(10)));


   if u(11) <= .01          % set output to zero
        set(hndls(14),'String',num2str(0.00));
   else
        set(hndls(14),'String',num2str(u(11)));
   end
                    % output the speed to analog display
        y2=[1.5 1.5+1.1*sin((2*pi*u(9)/60))];
          x2=[10.5 10.5-1.1*cos((2*pi*u(9)/60))];
          set(hndls(15),'XData',x2,'YData',y2);
          set(hndls(16),'XData',10.5,'YData',1.5);
        set(hndls(17),'String',num2str(u(12)));
        drawnow;
   end
sys=[u;x(13)];        % initialize system parameters

elseif flag == 0      % Initialize the figure "display" window
```

```matlab
      sys=[0;12+1;0;12;0;0];
   Figures = get(0,'Chil');
yesfig = 0;
 for i = 1:length(Figures)
   if strcmp(get(Figures(i), 'Name'), 'Output')
     yesfig = 1;
     whichone = i;
   end
 end
   if yesfig == 1
     curhndl = Figures(whichone);
     x0=[Inf;0;0;0;0;0;0;0;0;0;0;0;curhndl];
     set(curhndl,'Userdata',HNDL1);
   else
    curhndl = anim1(figname1);        % draw the speed dial
      cir = pi:-.035:-pi;
       for i=1:length(cir)
         x1(i) = 10.5 + 1.2*cos(cir(i));
         y1(i) = 1.5 + 1.2*sin(cir(i));
       end
       for i=1:length(cir)/15
         x1a(i) = x1(i*length(cir)/12);
         y1a(i) = y1(i*length(cir)/12);
       end
    x2=[10.5 9.4];
    y2=[1.5 1.5];
    hold on          % everything below initializes the display window
    plot(x1,y1,'r'),plot(x1a,y1a,'.w','MarkerSize',11);
    axis([0 12 0 3.8]);
    axis off
    text(8.9,1.55,'0','FontName','Times New Roman','FontSize',[10])
    text(10.3,3.,'15','FontName','Times New Roman','FontSize',[10])
    text(11.9,1.55,'30','FontName','Times New Roman','FontSize',[10])
    text(10.4,0,'45','FontName','Times New Roman','FontSize',[10])
    text(10.1,3.5,'Mph','FontName','Times New Roman','FontSize',[10])
    text(3.35,3,'Array Current (amps):','FontName','Times New Roman','FontSize',[10])
    text(3.95,2.5,'Run time (secs):','FontName','Times New Roman','FontSize',[10])
    text(3.72,2,'Odometer (miles):','FontName','Times New Roman','FontSize',[10])
    text(3.5,1.5,'Speedometer (mph):','FontName','Times New Roman','FontSize',[10]);
    text(-.1,3,' Motor Current (amps):','FontName','Times New Roman','FontSize',[10])
    text(.92,2,'Ahrs Used:','FontName','Times New Roman','FontSize',[10])
    text(1,1.5,'SOC (%):','FontName','Times New Roman','FontSize',[10])
    text(.05,.5,' Vehicle Load (N-m):','FontName','Times New Roman','FontSize',[10]);
    text(7.3,3,'  RPM:','FontName','Times New Roman','FontSize',[10]);
    text(6.4,2.5,'Hill Grade (%):','FontName','Times New Roman','FontSize',[10]);
```

```matlab
      text(-.1,1,'Battery Voltage (volts):','FontName','Times New Roman','FontSize',[10]);
      text(3.35,.5,'Motor Efficiency (%):','FontName','Times New Roman','FontSize',[10]);
      HNDL1(1)=text(2.2,3,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(2)=text(2.2,2,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(3)=text(2.2,1.5,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(4)=text(2.2,.5,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(5)=text(5.6,3,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(6)=plot(1,2.5,'.','MarkerSize',30,'EraseMode','background');
      HNDL1(7)=text(1.2,2.5,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(8)=text(5.6,2.5,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(9)=text(5.6,2,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(10)=text(5.6,1.5,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(11)=text(4,1,' ','Color',[1 0 0],'EraseMode','background','FontName','Times
New Roman','FontSize',[10]);
      HNDL1(12)=text(2.2,1,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(13)=text(5.6,.5,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(14)=text(8,3,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      HNDL1(15)=plot(x2,y2,'c','LineWidth',1.5,'EraseMode','background');
      HNDL1(16)=plot(10.5,1.5,'.','MarkerSize',20,'EraseMode','background');
      HNDL1(17)=text(8,2.5,' ','EraseMode','background','FontName','Times New
Roman','FontSize',[10]);
      hold off
      x0=[Inf;0;0;0;0;0;0;0;0;0;0;0;curhndl];
      set(curhndl,'UserData',HNDL1);
   end
else
   sys=[];
end;
```

G. FUNCTION ANIM1.M TO INITIALIZE ON-LINE DISPLAY WINDOW

```matlab
function [curhndl] = anim1(figname1);
%anim1 Initializes a figure for current animations.
```

% Steven Roerig. 25 July 1994, for SPEV


```
    % Now initialize the whole figure...
     position=[5 5 825 250];
     curhndl=figure( ...
        'Name',figname1, ...
        'NumberTitle','off', ...
           'BackingStore','off', ...
           'Position',position);
     axes( ...
        'Units','normalized', ...
        'Position',[.05 .1 .9 .75], ...
        'Visible','on', ...
           'DrawMode','fast');
```



```
cla reset
axis off
```


H. FUNCTION PLOTS.M TO GENERATE FINAL OUTPUT PLOTS

```
% This routine generates the plots for specfice outputs as
% well as averages for the entire simulation.

%   Plot the miles per hour and the total force on the vehicle
%
figure('Position',[0 340 300 300])
subplot(2,1,1),plot(miles,mph),grid
title('miles per hour')
subplot(2,1,2),plot(miles,Ftot),grid,xlabel('miles driven')
title('Total force on the vehicle (N)')

%   Plot the current drawn by the motor and the total amp hours used
%
figure('Position',[0 0 300 300])
subplot(2,1,1),plot(miles,iq),grid,
title('current drawn by motor')
subplot(2,1,2),plot(miles,Ahours),grid,xlabel('miles driven')
title('Amp-hours used')

%   Plot the power delivered to the drive wheel from the motor and
%   the power required at the rear wheel as a function of the forces
%
figure('Position',[340 0 300 300])
```

```
subplot(2,1,1),plot(miles,pwrdel),grid,
title('Power delivered to the rear wheels')
subplot(2,1,2),plot(miles,pwrreq./((datamat(1,15)/100)*effout)),grid,xlabel('miles driven')
title('Power required at the rear wheels')


%   Plot the efficiency of the motor for the simulation
%
figure('Position',[340 340 400 300])
subplot(2,1,1),plot(miles,effout),grid,
title('Motor efficiency')


%   Display the averages beyond the transient system startup
%
subplot(2,1,2)
axis([0 2 0 2]),axis off
text(.22,1.5,'Average Speed:
','FontSize',[9]),text(.8,1.5,num2str(mean(mph(100:length(mph)))),'FontSize',[9])
text(1,1.5,'mph','FontSize',[9])
text(0,1,'Average Current Drawn:
','FontSize',[9]),text(.8,1,num2str(mean(iq(100:length(iq)))),'FontSize',[9])
text(1,1,'Amps','FontSize',[9])
text(-.05,.5,'Average Motor Efficiency:
','FontSize',[9]),text(.8,.5,num2str(mean(effout(100:length(effout)))),'FontSize',[9])
text(1,.5,'%','FontSize',[9])
text(1.3,1.5,'Average Load:
','FontSize',[9]),text(1.8,1.5,num2str(mean(Tl2(100:length(Tl2)))),'FontSize',[9])
text(2,1.5,'Nm','FontSize',[9])
text(1.34,1,'Miles Driven:
','FontSize',[9]),text(1.8,1,num2str(miles(length(miles))),'FontSize',[9])
text(2,1,'miles','FontSize',[9])
text(1.25,.5,'Total Ahrs used:
','FontSize',[9]),text(1.8,.5,num2str(mean(Ahours(100:length(Ahours)))),'FontSize',[9])
text(2,.5,'Ahrs','FontSize',[9])
```

LIST OF REFERENCES

1.   Kyle, C.R., *Racing with the Sun, The 1990 World Solar Challenge*, Society of Automotive Engineers, Inc., Warrendale, PA, 1991.

2.   MATLAB, High-Performance Numeric Computation and Visualization Software, SIMULINK, Dynamic System Simulation Software, The MathWorks, Inc., Natick, Mass., 1994.

3.   Lebaric, J.E. and Manche, A.A., *"Design of Curved Solar Arrays for Solar-Powered Vehicles,"* SAE Technical Paper Series #911636, Rose-Hulman Institute of Technology, August 1991.

4.   Unnewehr, L.E. and Nasar, S.A., *Electric Vehicle Technology*, John Wiley and Sons, New York, NY, 1982.

5.   Krause, P.C. and O. Wasynczuk, *Electro-Mechanical Motion Devices*, McGraw-Hill, New York, NY, 1989.

6.   Krause, P.C., O. Wasynczuk, and Sudhoff, S.D., *Analysis of Electric Machinery*, McGraw-Hill, New York, NY, 1986.

7.   CALSOL Technical Report, University of California at Berkeley Solar Vehicle Race Team, 1991.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center   2
    Cameron Station
    Alexandria, Virginia   22304-6145

2. Library, Code 52   2
    Naval Postgraduate School
    Monterey, California   93943-5101

3. Chairman, Code EC   1
    Department of Electrical and Computer Engineering
    Naval Postgraduate School
    Monterey, California   93943-5121

4. Professor Jovan E. Lebaric, Code EC/Jl   1
    Department of Electrical and Computer Engineering
    Naval Postgraduate School
    Monterey, California   93943-5121

5. Professor Robert W. Ashton, Code EC/Ah   1
    Department of Electrical and Computer Engineering
    Naval Postgraduate School
    Monterey, California   93943-5121

6. LT. Steven J. Roerig   2
    4398 Barbara Way, N.E.
    Salem, Oregon 97305